MS-DOS 3.1
Reference

Rev. A

Important Software Diskette Information

For your own protection, do not use this product until you have made a backup copy of your software diskette(s). The backup procedure is described in the user's guide for your computer.

Please read the DISKID file on your new software diskette. DISKID contains important information including:

▶ The part number of the diskette assembly.

▶ The software library disk number (for internal use only).

▶ The date of the DISKID file.

▶ A list of files on the diskette, with version number, date, and description for each one.

▶ Configuration information (when applicable).

▶ Notes giving special instructions for using the product.

▶ Information not contained in the current manual, including updates, any known bugs, additions, and deletions.

To read the DISKID file onscreen, follow these steps:

1. Load the operating system.

2. Remove your system diskette and insert your new software diskette.

3. Enter

   type diskid ↵

4. The contents of the DISKID file is displayed on the screen. If the file is large (more than 24 lines), the screen display will scroll. Type Ctrl-S to freeze the screen display; type Ctrl-S again to continue scrolling.

Contents

Appendixes

Figures

Tables

Chapters

1

2

3

4

5

6

7

A

B

C

D

Preface

This manual is organized as follows:

▶ Chapter 1 discusses the structure of MS-DOS, how the system uses memory and disk space, how to load MS-DOS, and other introductory topics.

▶ Chapter 2 describes the types of disk files: system files and utilities, data/text files, and batch files. This chapter also describes how to use device files, and gives the rules for naming files and for using wildcard characters.

▶ Chapter 3 discusses the hierarchical (tree-structured) MS-DOS directory system, and tells how to use subdirectory names, create and delete directories, move between directories, and find files in the directory system.

▶ Chapter 4 explains how to use EDLIN, a text editor used mainly to edit source files.

▶ Chapter 5 tells how to edit MS-DOS command lines, how to redirect input/output, and how to "filter" and "pipe" data.

▶ Chapter 6 describes all the MS-DOS commands—the internal and external commands, batch commands, and system configuration commands. This chapter describes the syntax for each command, the parameters used with each command, and examples of usage.

▶ Chapter 7 lists and describes messages the system might display during bootup and during normal operations.

▶ Appendix A is an ASCII-to-hexadecimal conversion table.

▶ Appendix B lists the device input/output errors.

▶ Appendix C describes the ANSI escape sequences.

▶ Appendix D contains information on the DEBUG program. All the DEBUG commands are listed alphabetically with a description.

Manual Conventions

This manual uses the following conventions:

▶ When commands are explained for the first time, the command format is set off in boldface, such as:

DO filename /switch↵

— The parts of the command that you type exactly as shown (such as the command name) are all uppercase.

— Parts of the command that you can vary or omit (such as file-names and drive names) are all lowercase.

— When you type a command, you can use uppercase or lowercase for any command elements.

▶ In examples and command formats, pressing the Enter key is shown as ↵, and pressing the Spacebar is shown as (sp). Function keys are shown as (Fx); for example, (F2) indicates function key F2.

▶ Some examples show the screen, including what you type and the system's response. In these screen examples, what you type is under-lined and all-lowercase. For example:

```
A>print myfile↵
```

▶ In the text, names of commands, files, and programs appear in all-uppercase. Examples are COPY and TEXT.DOC.

▶ In text, "Ctrl-" represents the Ctrl key. For example, Ctrl-C repre-sents pressing the Ctrl key while you type C. In screen examples, the Ctrl key is shown as a caret (^), such as ^C.

Introduction

MS-DOS (Disk Operating System) is a group of programs that:

▶ Manage files and directories

▶ Process commands

▶ Control application programs designed for MS-DOS

▶ Control the operation of your keyboard, disk drive(s), and printer(s)

VICTOR includes several extensions with this version of MS-DOS. These extensions include an internal HISTORY command and several utilities that are frequently used with the UNIX™ operating system.

The first part of this chapter gives short descriptions of the system files and utilities that make up MS-DOS. Also included is a discussion of how MS-DOS uses memory (RAM) and disk space. The last part of the chapter describes the system configuration, and the procedures for loading MS-DOS and for entering the date and time.

1.1   MS-DOS System Structure

MS-DOS consists of the following files and programs:

▶ System Files

COMMAND.COM is a command processor that accepts commands from the keyboard and runs the programs that process them.

CONFIG.SYS is a group of system initialization routines and the commands for changing the system configuration (see Section 6.3).

MSDOS.SYS and IO.SYS are the actual operating system programs that manage files and application programs. MSDOS.SYS and IO.SYS are hidden system files.

▶ Special Files

ANSI.SYS is an optional loadable console device driver.

VDISK.SYS loads a virtual disk into memory.

Note: Your system diskette may also include an AUTOEXEC.BAT file or a CONFIG.BAT file. See Chapter 2 for information on the AUTOEXEC.BAT file and the CONFIG.BAT file.

▶ Command Files (Utility Programs)

ASSIGN assigns a logical designator to a drive.

ATTRIB sets or resets the read-only attribute of a file.

BACKUP copies files from the fixed disk to diskettes.

BASICA is the VBASICA interpreter.

CHKDSK displays the amount of disk space occupied by files and tells you how much space is available for storage. CHKDSK also scans the directory for errors and can correct allocation problems.

COMMAND loads the command processor (COMMAND.COM) into memory.

COMP compares the contents of one file to the contents of another file.

CONCAT concatenates (combines) files, displays the files onscreen, or sends them to a file.

DEBUG is a debugging program.

DISKCOMP compares the files of two diskettes.

DISKCOPY copies the contents of one diskette onto another.

EDLIN is the MS-DOS line editor program.

EXE2BIN converts executable files (.EXE) to binary (.COM) format.

FGREP searches through files or directories for a specified expression and prints the names of the files and the lines containing the expression.

FIND searches for a specified string of text.

FORMAT prepares a diskette to receive MS-DOS files.

GRAFTABL loads a table of additional character data for the color/graphics adapter used in graphics mode.

GRAPHICS sets up a graphics printer to allow printing of graphics display screens.

JOIN joins a drive to another drive or a directory.

KEYBFR loads the French keyboard program into memory.

KEYBGR loads the German keyboard program into memory.

KEYBSV loads the Swedish keyboard program into memory.

KEYBUK loads the British keyboard program into memory.

LABEL creates, changes, or deletes a volume label.

LINK is the MS-DOS linker. The LINK.DOC file on your system diskette describes how to use LINK.

LS alphabetically lists a directory, or lists selected parts of a directory according to specified constraints.

MODE sets up the display, printer ports, and asynchronous communications adapters.

MORE displays the contents of files one screen at a time.

MV moves files (makes a copy of a file and deletes the original), and renames files.

PRINT queues a list of files for printing.

RECOVER recovers lost data from a file or from a whole disk.

RESTORE copies files from backup diskettes to the fixed disk.

SEARCH selects files from a given set of files according to matching constraints, and performs a specified action on the matching files.

SELECT creates a new operating system diskette configured with a specific keyboard layout and time and date format.

SHARE sets up file and record sharing calls.

SORT sorts specific lines of data from a file or arranges the file in ascending or descending order.

SUBST substitutes a different drive specifier for a drive or path.

SYS copies the hidden system files from one diskette or fixed disk volume to another.

TAIL displays the final lines of a file.

TREE lists all directory paths on a drive.

UNIQ searches SORTed files for duplicate lines and deletes the redundancies.

WC counts words and/or lines in a file or a group of files.

Section 6.4 contains detailed descriptions of all the MS-DOS commands alphabetically. Refer to Section 6.4 for explanations of commands as you read the earlier chapters.

The hidden system files MSDOS.SYS and IO.SYS contain programs that run automatically when MS-DOS loads into memory. The files COMMAND.COM, MSDOS.SYS, and IO.SYS control the system resources. The relationship between these files and system resources is shown in Figure 1-1.
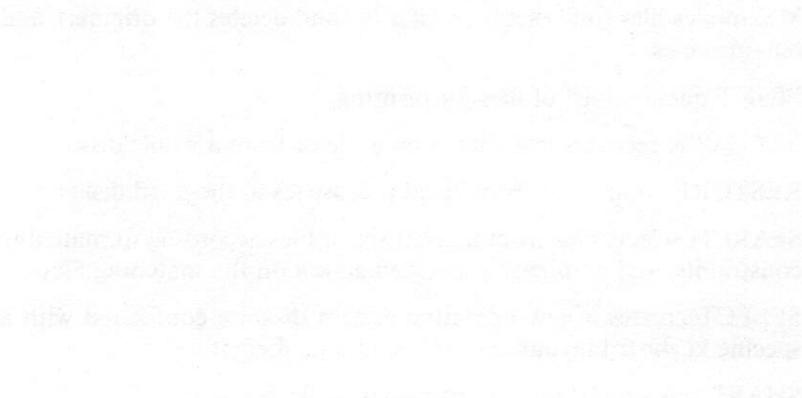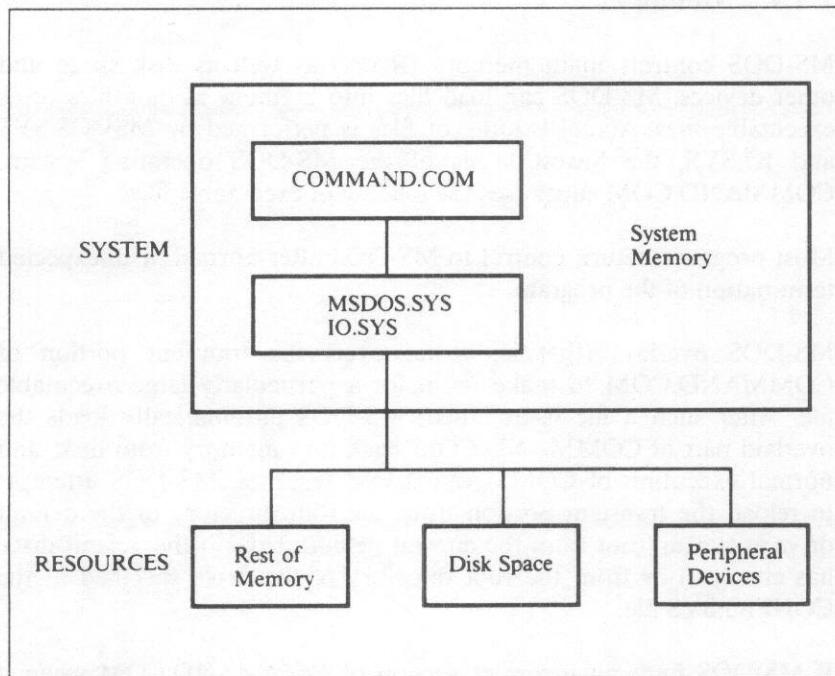
Figure 1-1: MS-DOS and Its Resources

The system's resources include its internal memory (RAM) and external storage (disk space), plus connections to the screen, keyboard, printer(s), other peripheral devices, and a network configuration, if any.

Memory and the MS-DOS file system can be thought of as the external organization of system resources. MS-DOS supports "device independent I/O." This means that you can treat devices (such as your printer, keyboard, and screen) as if they are files when you do input/output processing (see Section 2.4).

The system's memory and its disk space are described in Sections 1.1.1 and 1.1.2.

1.1.1 Memory

MS-DOS controls main memory (RAM) as well as disk space and other devices. MS-DOS can load files into memory as data files or as executable files. Actual loading of files is performed by MSDOS.SYS and IO.SYS, the lowest levels of the MS-DOS operating system. COMMAND.COM supervises the loading of executable files.

Most programs return control to MS-DOS after normal or unexpected termination of the program.

MS-DOS overlays (that is, writes over) the transient portion of COMMAND.COM to make room for a particularly large executable file. After such a file is executed, MS-DOS automatically loads the overlaid part of COMMAND.COM back into memory from disk, and normal execution of COMMAND.COM resumes. MS-DOS attempts to reload the transient portion from the root directory of the default drive at startup (not from the current default drive, if the default drive has changed) or from the root directory of the drive specified in the CONFIG.SYS file.

If MS-DOS finds an incorrect version of COMMAND.COM when it attempts to load the overlaid section, it displays a message such as:

```
Invalid COMMAND.COM
  COMMAND.COM expected at X:\COMMAND.COM
  <CR> for retry, or new COMMAND.COM path =
```

where drive X is the original default drive. Insert the system diskette you used to load MS-DOS (or any diskette containing the version of COMMAND.COM that was originally loaded). To use another drive or path for reloading COMMAND.COM, type the name of that drive or path. Now MS-DOS will use that drive or path each time it must reload COMMAND.COM during this operating session.

1.1.2   Disk Space

In MS-DOS, disk space is divided into four parts:

▶ The reserved sectors contain information used each time MS-DOS is loaded.

▶ The directory contains information about each file on a given diskette. This information includes the file's complete filename, its size, and the time and date of its last modification.

▶ The file allocation table (FAT) contains location information for the data making up each file on a diskette.

▶ Files occupy the majority of disk space. An individual file does not necessarily reside in contiguous areas on the diskette; its contents might be "scattered" on the diskette so disk space is not wasted.

1.2   System Configuration

With MS-DOS your system configuration can include these peripheral devices:

▶ One parallel printer

▶ One serial printer

▶ A MODEM option (telephone hookup)

To set up your operating system for your particular configuration, refer to your user's guide and to the discussions of the system configuration commands and CTTY in Chapter 6 of this manual.

1.3   Loading MS-DOS

Before you start working with your system, you should make a backup of your MS-DOS system diskette using the DISKCOPY program.

To load MS-DOS, insert the system diskette and reset the system by pressing the Alt, Ctrl, and Del keys simultaneously.

Once MS-DOS is in main memory, it prompts you to enter the date:

```
Enter new date:_
```

MS-DOS keeps track of the date and time. If you enter the date and time when you load MS-DOS, it records the current date and time on each file that you create or change.

To set the date, follow these guidelines:

▶ The date format is determined by the COUNTRY code in the CONFIG.SYS file (see Section 6.3). Enter the date in the appropriate format for your operating system:

— month/day/year or month-day-year (US)

— day-month-year or day/month/year (Europe, except Sweden)

— year/month/day or year-month-day (Sweden and Japan)

▶ For the month, enter a number from 1 to 12. For the day, enter a number from 1 to 31. For the year, enter a number from 80 to 99 or from 1980 to 2099. Then press Enter to submit your entry to MS-DOS.

▶ Use the Backspace key to correct mistakes.

▶ Do not type the name of the day because MS-DOS computes the day automatically.

▶ If you do not want to type in a date, press the Enter key. Then MS-DOS uses the system creation date or the last valid date setting as the current date.

For example, to enter the date February 25, 1986, for an American operating system. type

2/25/86

or

2-25-86

If the date you enter is not in the correct format or contains a number outside the valid ranges (such as 13 for the month), MS-DOS displays

```
Invalid date
Enter new date:_
```

After MS-DOS accepts your entry for the date, it displays its current time setting and prompts you to reset the time:

```
Enter new time:_
```

To enter the time, you must use a 24-hour clock. For example, 1 PM is 13:00, and 11 PM is 23:00. When MS-DOS displays the time, it shows the hour, minute, second, and half second (0:00:00.0).

To change the time:

▶ Use the numbers 0–23 for the hour and 0–59 for the minutes. You do not have to make an entry for minutes or seconds. Do not type the half second.

▶ Separate the time with colons (hour:minute:second).

▶ Use the Backspace key to correct mistakes.

▶ Press the Enter key when you want MS-DOS to accept your entry.

If you type an invalid time, after you press Enter MS-DOS displays

```
Invalid time
Enter new time:_
```

Retype the time.

After MS-DOS accepts your entry for the time, the MS-DOS command-line prompt and the cursor appear on your screen. By default, the prompt displays the name of the drive from which MS-DOS was loaded. For example, if you load MS-DOS from drive A, the prompt is

```
A>_
```

The command-line prompt indicates that MS-DOS is in main memory and is ready for you to enter an MS-DOS command from the keyboard. You can change the prompt to be any letter(s), word, or phrase with the PROMPT command (see Section 6.4 for more information).

You can check and change the date or time while you are using MS-DOS. Type the word **date** or **time** and press Enter. MS-DOS shows the date or time and asks you to reset it. Press Enter to leave it as is, or follow the rules for entering the date or time given in this section.

Note: The SHELL command in the CONFIG.SYS file can instruct MS-DOS to load the command processor without prompting you for the date and time of bootup. See COMMAND in Section 6.4.

2

The MS-DOS File System

This chapter describes the MS-DOS file system, including:

▶ Types of files you can use with MS-DOS

▶ Conventions for naming files

▶ Rules for using wildcard characters

▶ How to create and run batch files

▶ The function of the AUTOEXEC.BAT and CONFIG.BAT files

▶ How to use peripheral devices (device files) for input/output

MS-DOS stores and processes data as files. A file is made up of one or more related characters. The capacity of a file is limited only by the data storage capacity of a diskette or fixed disk volume.

MS-DOS uses files as locations for writing or reading data. A file's location can be a diskette, or a peripheral device, such as the keyboard, screen, or printer. The data in most files has been input from the keyboard.

2.1   Types of Files

MS-DOS recognizes these types of files:

▶ System files

▶ Command files (utility programs)

▶ Data/text files

▶ Batch files

▶ Source files

System files are predefined programs that run the computer and its devices, control other programs loaded into memory from disk such as word processing programs, and process data entered from the keyboard.

A **command file** is a predefined program on disk. You run the program by typing the filename (without the extension), followed by optional parameters. The terms "external command" and "utility program" both refer to command files. Utilities or external commands perform basic file and diskette management tasks (such as formatting and copying diskettes) that are necessary for every application. Each utility program is stored on disk with a filename extension of .COM or .EXE.

A **data/text file** is a collection of alphanumeric characters that can be in document form, such as a memo. It can also be a list of data, such as a mail list, payroll statement, or the output of an application program, like VBASICA.

A **batch file** is a sequence of MS-DOS commands. When you load a batch file, it performs a specific task or tasks (see Section 2.5).

A **source file** contains English-like program language statements that you create. Your computer uses a memory-resident translation program called a compiler to put each source statement line into a format that the machine can process.

2.2 Filename Conventions

The filename conventions described in this section are rules for speci-
fying the name of a file, the filename extension, and the drive name. If
you follow clear and simple file-naming conventions, you will be able
to easily identify the contents of your files.

2.2.1 Filenames

Names for MS-DOS files can be one to eight alphanumeric characters.
You can use these characters:

▶ The letters A through Z (uppercase or lowercase)

▶ The numbers 0 through 9

▶ The following special characters:

   ! @ # $ % & ( ) - ' ^ { } ~ `

If you use any other characters, MS-DOS displays the message "Invalid
filename." You also cannot give your files the device names listed in
Table 2-2.

2.2.2 Filename Extensions

You can add an optional filename extension to a filename. The exten-
sion is one to three alphanumeric characters preceded by a period. The
valid characters for extensions are the same as for filenames.

You can use extensions to identify types of files. Table 2-1 lists some
conventional MS-DOS file extensions and their meanings. Extensions
other than those given in Table 2-1 or defined by your application pro-
gram can be assigned as you wish. For example, you can use your ini-
tials as the extension for the files you create, or the extension .TXT for
text files, or .MEM for memos.

If you specify more than three characters for the extension, MS-DOS accepts only the first three characters. If a data/text file has a filename extension, you must include the extension with the filename when you make a request for the file. To run a command file or batch file, you omit the extension (.COM, .EXE, or .BAT) when you type the filename.

Table 2-1: MS-DOS File Extensions

FILE EXTENSION	MS-DOS INTERPRETATION
.ASM	8086 assembler language source code
.BAK	Backup file created by application
.BAS	BASIC source code (VBASICA)
.BAT	Batch command file
.COB	COBOL source code
.COM	Executable command file
.CRF	Cross-reference file
.DAT	Data file
.EXE	Relocatable executable file
.FOR	FORTRAN source code
.INT	Intermediate compiled code
.LIB	Library file
.LST	Listing of compilation or assembly
.MAP	Memory map from linker program
.OBJ	Relocatable object code module
.OVR	Overlay module
.PAS	Pascal source file
.PLM	PL/M source code
.PRN	Listing of compilation or assembly
.REF	Cross-reference listing
.$$$	Temporary system-generated file

2.2.3 Drive Names

The drive name is the letter name of the disk drive where the file is currently located. Place a colon between the drive name and the filename itself. For example, to indicate that the file MAIL.LST is on drive B, type

b:mail.lst

If you enter a filename without a drive name, MS-DOS searches for the file on the default drive.

To fully identify a file for MS-DOS, you might need to include the name of the subdirectory that contains the file. See Chapter 3 for information on using directory names. This is the format for specifying a complete filename:

d:filename.ext

The following description summarizes the characteristics of each part of the filename.

d:

▶ Is one letter

▶ Is followed by a colon

▶ Does not need to be specified when the file is on the default drive

filename

▶ Contains up to 8 characters

▶ Does not use the symbols " / [ ] + = , ; : | < > \ (sp)

.ext

▶ Is preceded by a period (.)

▶ Contains up to 3 characters

▶ Does not use the symbols " / [ ] + = , ; : | < > \ (sp)

▶ Is optional except for some of the extensions in Table 2-1

2.3   Filename Wildcard Characters

The ? and * wildcard characters replace existing characters in a file-
name. You can use wildcard characters to match a specified portion of
a filename with existing filenames.

Wildcard characters can be used with these commands:

BACKUP   PRINT
COPY   REN
DEL   RESTORE
DIR

The ? wildcard character used in a filename with an MS-DOS com-
mand matches any single existing character (or none) in that position.
For example, suppose a sample directory in the default drive contains
several similar filenames, such as:

CHAPT1
CHAPT2
CHAPT3

If you enter

dir chapt?↵

MS-DOS displays only the filenames that match the first five char-
acters of the filename in the command. The sixth character can be any
character, as indicated by the ?.

The * wildcard character replaces any existing character in that posi-
tion and all subsequent characters. A matching filename can be 2 to 8
characters long, and the other characters in the filename can be any
character.

2-6

For example, using the sample directory, if you enter the following command MS-DOS displays all the filenames that start with CH:

dir ch*⌄

MS-DOS also displays any other filenames in the directory that begin with CH, such as CHARLES.MEM, CHART.DAT, or CHECKS.

You can also use the * to display all the files ending with a particular extension. For example, if you enter

dir *.bat⌄

MS-DOS displays the names of all the batch files (indicated by .BAT) in the current directory.

You can specify *.* in a command to indicate all the filenames in a directory. For example, using *.* with the DEL command deletes all the files from the directory. When you specify DEL *.*, MS-DOS asks if you are sure before it deletes the files. Make sure that you have backed up any files you need.

You can use ? and * together in a command. For example, suppose your directory contains these files:

83JOBS.PUB
82JOBS.PUB
83JOBS.ACT

Then if you enter

dir 8?jobs.p*⌄

MS-DOS displays the filenames 83JOBS.PUB and 82JOBS.PUB. 83JOBS.ACT is not displayed, because .A does not match .P.

2.4 Device Names

MS-DOS can send data to or receive data from a device. MS-DOS interprets device names as the names of peripheral devices. You can use these device names as parameters in MS-DOS commands to refer to devices as if they were files. A device name never has a file extension or drive name because device names are not disk files, but represent input/output devices. "Logical device" is another term for a device name.

The device names listed in Table 2-2 include the logical device names for the standard MS-DOS device drivers (programs that control the peripheral devices connected to your computer). Device drivers are contained on the system diskette. At bootup, drivers are loaded by the CONFIG.SYS file, and the drivers' names (logical device names) are displayed on the screen.

Table 2-2: MS-DOS Device Names

NAME	DEVICE TYPE
AUX	An input or output auxiliary device.
COM1	The name for the device driver for serial port 1.
COM2	The name for the device driver for serial port 2.
CON	Keyboard input to the CPU, and CPU output to the screen.
LPT1	The name for the device driver for the first parallel port.
LPT2, LPT3	The names for the device drivers for the second and third parallel ports.
NUL	A nonexistent device for a command that requires a filename. Use this device when you do not want to create a file. You can also use this device to get rid of unwanted console output.
PRN	Same as LPT1.

For example, you can copy data from the CON device file (the keyboard) to a file on disk called PHN.LST:

copy con phn.lst↵

The next lines you type after the command are temporarily stored in memory until you press Ctrl-Z and the Enter key. Then the text you have entered is stored in the file PHN.LST. Data to or from CON (or any device) must always be ended with Ctrl-Z, followed by Enter.

You can also redirect output to a device by using the > symbol (see Chapter 5). For example, this command sends the directory to the printer (PRN):

dir > prn↵

When you specify a device name, such as PRN, be sure that the correct device type is assigned to PRN.

2.5 Batch Files

A batch file is a sequence of MS-DOS commands that perform a specific task. You can create batch files with the line editor EDLIN, with the COPY command (described in Chapter 6), or with any word processing program. You must give batch files the extension .BAT. To run the sequence of commands stored in a batch file, type the name of the file without the extension.

Using batch files, you can execute several MS-DOS commands by entering a single batch command. Because batch files can perform many different functions, writing batch files is similar to programming, and the files themselves are like short programs "written in MS-DOS." This chapter gives the basic information you need to start using and writing batch files; there are examples of batch file usage throughout this manual.

The following example is a batch file named NEWDISK.BAT that contains a sequence of commands to display the directory of a formatted diskette and check the diskette for consistencies. It also displays remarks (lines beginning with REM) that describe the batch file.

REM This is a file to check formatted diskettes
REM It is named NEWDISK.BAT
PAUSE Insert diskette in drive B
DIR B:
CHKDSK B:

Once the NEWDISK.BAT file is saved on disk, you can run the two commands (DIR B: and CHKDSK B:) by typing the batch file name NEWDISK. PAUSE and REM are commands used only for batch files. These commands and the other commands used for batch files (FOR, IF, GOTO, ECHO, and SHIFT) are described in Section 6.4.

2.5.1   Batch File Conventions

Follow these rules when you use batch files:

▶ To run a batch file, enter the batch filename (without the .BAT extension). If the batch file contains replaceable parameters, follow the filename with the values you want to substitute for the parameters (see Section 2.5.2).

▶ If you press Ctrl-C while a batch file is being processed, MS-DOS displays

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

Type Y if you want MS-DOS to ignore the rest of the commands in the batch file and return you to the system prompt. Type N to stop the current command being processed; MS-DOS then processes the next command in the batch file.

▶ You can specify the name of another batch file as a command in the batch file. MS-DOS then invokes that batch file from the one being processed. MS-DOS does not return you to the first batch file, however, unless the first batch file invokes a new copy of COMMAND.COM which in turn has invoked the second batch file.

▶ If you invoke a batch file that changes directories, MS-DOS does not automatically return you to the directory from which you invoked the batch file.

2.5.2   Replaceable Parameters

Replaceable parameters are "dummy values" in a batch file. They are numbers preceded by a percent sign, such as %1 and %2. When you run a batch file, you include the value(s) you want to use in place of the replaceable parameters in the file.

You can specify ten replaceable parameters in a batch file (%0 to %9). %0 is always replaced by the drive name (if one is designated) and the name of the batch file. %1 is replaced by the first value following the filename in the command you enter to run the batch file. %2 is replaced by the second value, and so on. You can specify more than ten replaceable parameters by using the SHIFT command.

For example, you can use the COPY CON command to create a phone list file:

copy con phn.lst⏎
Mary Smith (415) 497-0110⏎
Engineering (408) 438-6680⏎
George Jones (916) 555-0900⏎
Marketing (408) 438-7000⏎
^Z⏎

Then you can create a batch file named LOOKING that uses the FIND command to look through PHN.LST for a dummy value (the replaceable parameter %1):

```
copy con looking.bat↵
find "%1" phn.lst↵
^Z↵
```

When you load the batch file, include the value or name that you want to substitute for the replaceable parameter in the batch file. For example, if you specify

```
looking 408↵
```

MS-DOS replaces the parameter %1 with 408. Then MS-DOS processes the LOOKING batch file using this value and displays the command from the file:

```
A>find "408" phn.lst
```

The FIND command then displays all the lines from the PHN.LST file that contain 408:

```
---------- phn.lst
Engineering (408) 438-6680
Marketing (408) 438-7000
```

2.5.3 The AUTOEXEC.BAT and CONFIG.BAT Files

AUTOEXEC.BAT is the name of a batch file that MS-DOS always looks for and processes after it loads into memory. If there is no AUTOEXEC.BAT file, MS-DOS prompts for the date and time and displays its command prompt. If there is an AUTOEXEC.BAT file, MS-DOS displays the time and date prompts only if the DATE and TIME commands (described in Section 6.4) are included in the AUTOEXEC.BAT file.

Create an AUTOEXEC.BAT file if you want a command or sequence of commands to operate each time MS-DOS loads. For example, suppose your principal application is VBASICA. You can use COPY CON to create an AUTOEXEC.BAT file that takes you straight into VBASICA when you boot up:

copy con autoexec.bat ↵
basica ↵
^Z ↵

The command BASICA tells MS-DOS to load the VBASICA language into memory. If this AUTOEXEC.BAT file and the command file are on your system diskette, then each time MS-DOS loads into memory, it displays the first command from the file:

```
A>basica
```

VBASICA loads into memory and displays its command prompt (Ok). To return to MS-DOS, use the VBASICA command SYSTEM to exit to the operating system.

If you create an AUTOEXEC.BAT file and want to process it immediately, either reload MS-DOS or specify

autoexec ↵

The CONFIG.BAT file, which is a VICTOR extension to MS-DOS, differs from an AUTOEXEC.BAT file only in that it does not cause the operating system to bypass the DATE and TIME prompts. You can invoke batch file commands from either of these files. If you have both a CONFIG.BAT file and an AUTOEXEC.BAT file in the root directory of the boot volume or drive, the CONFIG.BAT file is invoked first.

3

The MS-DOS Subdirectory System

MS-DOS allows you to create and use a hierarchical subdirectory system. This chapter describes the subdirectory system and how to work with it. The first topic is a general introduction, followed by the rules for using subdirectory names to locate files. Then the chapter describes the MS-DOS commands used with subdirectories:

▶ DIR displays the current directory.

▶ MKDIR creates a subdirectory.

▶ CHDIR changes from one directory to another.

▶ RMDIR deletes an empty subdirectory.

A **directory** is a list of the files on a disk. The list includes the filenames, the file size, and the date and time each file was created. MS-DOS enters the filename and date and time in the directory each time you create or update a file.

MS-DOS has a main directory to which you can add subdirectories. Subdirectories are hierarchical, like a family tree or an organization chart. The main or "root" directory is created by the FORMAT program during a disk format. The root directory is the start of your subdirectory system; it contains the volume label and files and can also contain the names of subdirectories. Each subdirectory in turn can contain files and other subdirectories.

For example, you can create a subdirectory called WORD.DIR and place your word processing application program in that subdirectory. Then you can store all the text files you create in the word processing subdirectory, WORD.DIR, as shown in the following diagram.

```
         ┌──────────────┐
         │     Root     │
         │  Directory   │
         └──────────────┘
                │
         ┌──────────────┐
         │     Word     │
         │  Processing  │
         │   Program    │
         │   WORD.DIR   │
         └──────────────┘
                │
         ┌──────────────┐
         │  Text Files  │
         │  CHAP1.TXT   │
         │  SMITH.LET   │
         │  BUYER.MEM   │
         └──────────────┘
```
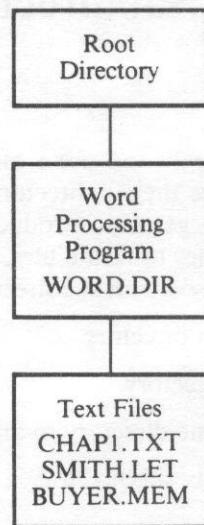
Similarly, you can create a different subdirectory for each application program, and group your files into the appropriate subdirectory:

```
                ┌──────────────┐
                │     Root     │
                │  Directory   │
                └──────────────┘
                       │
      ┌────────────────┼────────────────┐
┌──────────┐    ┌──────────────┐   ┌──────────┐
│ Business │    │     Word     │   │  Sales   │
│   Plan   │    │  Processing  │   │ Forecast │
│ BUS.DIR  │    │   Program    │   │SALES.DIR │
│          │    │   WORD.DIR   │   │          │
└──────────┘    └──────────────┘   └──────────┘
```
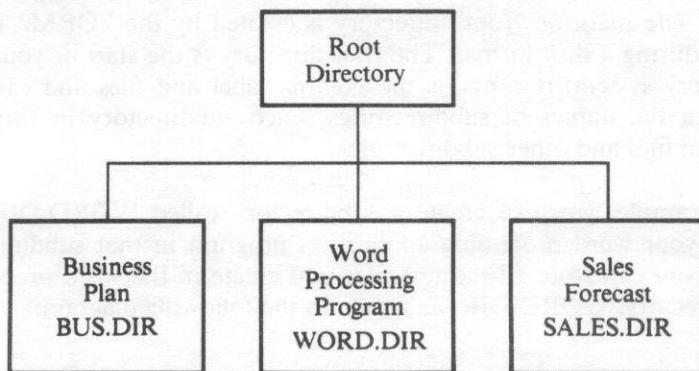
When you save a file, MS-DOS puts the filename in the directory you are currently using or in another directory that you specify. A file in one subdirectory can have the same name as a file in another directory because MS-DOS keeps each directory separate.

You can search for a file in one subdirectory rather than searching through many kinds of files in one large main directory. Section 3.1 describes how to specify the subdirectory name (or path) for a file.

3.1 Using Directory Names (Paths) to Locate Files

When you use hierarchical directories, you must tell MS-DOS where to find your files in the directory structure. In addition to the filename, you give the name of the subdirectory where the file is located.

Giving the subdirectory name means specifying a **path** for MS-DOS to follow through the directory hierarchy. A path or subdirectory name:

▶ Is a sequence of directory names, each separated from the previous one by a backslash (\).

▶ Leads MS-DOS through the directory hierarchy. Each \ leads to a lower subdirectory level.

▶ Can end with a filename or a directory name.

▶ Can go through only existing subdirectories.

▶ Must not exceed 63 characters.

The format for a path to a file or subdirectory is as follows:

[d:][[\]directoryname[\directoryname...]\][filename.ext]

d:
    is the name of the drive containing the file or directory you want. You can omit the drive name if the file or directory is on the default drive.

\

An initial backslash always indicates the root directory. If a path begins with a backslash, MS-DOS begins searching for the file or directory at the root directory. Otherwise, MS-DOS searches downward from the current directory.

directoryname

is the name of a subdirectory. If the first subdirectory listed is not preceded by a backslash, it is directly below the current directory. Each additional \directoryname indicates another subdirectory level below the root directory or below the current directory.

filename.ext

is the full filename and extension of the file you want. The filename is always last in the path. If the file is in the current directory, you do not need to specify a path with the filename.

MS-DOS recognizes three abbreviated pathnames:

▶ . is shorthand for the current directory. You can use the . notation with the DIR command to display the current subdirectory, but it is not necessary.

▶ .. is shorthand for the current directory's parent directory. MS-DOS uses the . and .. directories to create the hierarchical tree directory structure. You can use the .. notation with DIR, CHDIR, and DEL.

▶ \ is shorthand for the root directory. You can use the \ notation with the CHDIR command.

3.1.1 Examples of Paths

The following examples demonstrate different kinds of pathnames.

\WORD.DIR
\MGR1.NAM

> These are subdirectory names. The initial \ indicates the root directory. WORD.DIR and MGR1.NAM are first-level subdirectories (directly below the root directory).

\WORD.DIR\CHAP1.TXT
\MGR1.NAM\JOBS\CLERK

> These are full pathnames. In the first example, the file or subdirectory CHAP1.TXT is in the first-level subdirectory WORD.DIR. In the second example, the file or subdirectory CLERK is in the subdirectory JOBS under the first-level subdirectory MGR1.NAM.

WORD.DIR\DAVID
JOBS\BENEFITS

> These are relative pathnames that refer to the current directory level. In the first example, DAVID is a file or directory in the subdirectory WORD.DIR below the current directory. In the second example, the file or directory BENEFITS is in the JOBS subdirectory, which is listed in the current directory. If the current directory is the root directory, these pathnames are the same as specifying \WORD.DIR\DAVID and \JOBS\BENEFITS.

3.1.2 Working with Files in Subdirectories

You can use paths or subdirectory names with any of the MS-DOS commands that operate on filenames. This section gives several examples of using pathnames in MS-DOS commands.

You locate a file by specifying a path for MS-DOS to follow through the directory hierarchy. The filename is always the last name in the path. When the file is in the current directory, you can omit the path and give just the filename. To get to a file in the previous directory, you can use .. as the path.

Displaying Files Onscreen

You can display the contents of text files on your screen using the TYPE command. For example, to display the CHAP1.TXT file in the WORD.DIR subdirectory on the default drive, enter

type \word.dir\chap1.txt ↵

The first \ tells MS-DOS to start the search for the file at the root directory. MS-DOS finds the subdirectory WORD.DIR listed in the root directory. In WORD.DIR, MS-DOS finds the CHAP1.TXT file and displays the file on the screen.

If WORD.DIR is the current directory, you can specify the filename without a path, like this:

type chap1.txt ↵

If you include the drive name in a command, the drive name must precede the path. For example, this command displays CHAP1.TXT from the WORD.DIR subdirectory on drive B:

type b:\word.dir\chap1.txt ↵

Copying Files

You use the internal command COPY to copy files. To copy a file from one directory to another, you must specify the path to the file to be copied and the path to the directory where the file is to be copied. For example, to copy the CHAP1.TXT file from the WORD.DIR subdirectory on drive A into the root directory on drive B, and to rename the file INTRO, enter

    copy a:\word.dir\chap1.txt b:\intro ↵

Deleting Files

To delete files from a directory, use the internal command DEL (Delete). If the file you want to delete is in the current directory, you do not need to specify a path to the file. For example, if you are using the WORD.DIR directory, you can delete a file from that directory with this command:

    del chap1.txt ↵

If you are not in the WORD.DIR directory, you must include the path to the file you want to delete from WORD.DIR. For example, this command deletes CHAP1.TXT from the WORD.DIR subdirectory on drive B:

    del b:\word.dir\chap1.txt ↵

You can use the .. notation with DEL to refer to the previous directory. For example, this command deletes MYFILE from the parent directory of the current directory:

    del ..\myfile ↵

3.1.3 Paths and External Commands

External commands are programs stored in disk files. When you are working with more than one directory, you might want to put all the external commands in one directory to keep the other directories uncluttered. If you do this, you must be in the subdirectory containing the external command in order to load that command.

For example, suppose you store all the external commands in the BIN first-level subdirectory. To load and run the CHKDSK program, you must first move to the BIN subdirectory.

Instead of moving to the subdirectory, however, you can set up the path to the subdirectory with the PATH command. For this example, you would enter this PATH command:

path x:\bin ⏎

where x is the name of the default drive. Now MS-DOS searches the current directory and the BIN subdirectory whenever you specify an external command. You have to specify PATH only once in an operating session (see PATH in Section 6.4).

The PATH command searches only for .EXE, .COM, and .BAT files. Text files and other types of files cannot be accessed using the PATH command.

3.2 Listing a Directory

The DIR command, described in Section 6.4, lists the contents of a directory. You can list a directory onscreen or send it to a device. To list a subdirectory with DIR, you can use the path notation shown in Section 3.1.

For example, you can display the current directory on the default drive by entering

 dir ⏎

To display the parent directory (one level above the current directory), enter

 dir .. ⏎

If you are in the WORD.DIR subdirectory, entering DIR displays that subdirectory. To display the WORD.DIR subdirectory on the default drive (regardless of the directory you are in), enter

 dir \word.dir ⏎

The display for the WORD.DIR subdirectory might look like this:

```
Volume in drive A has no label
Directory of A:\word.dir

.            <DIR>            5-05-86   10:09a
..           <DIR>            5-05-86    2:15p
DAVID.L      <DIR>            5-20-86    3:40p
CHAP1.TXT              23345  6-15-86   11:13a
        4 File(s)    370688 bytes free
```

This DIR display gives the following information:

▶ No Volume ID was assigned to the disk during formatting.

▶ This directory is WORD.DIR, a first-level subdirectory.

▶ The single period (.) indicates the current directory. The . is displayed only for subdirectories, not for the root directory. MS-DOS automatically creates the . entry when you create a directory.

▶ Two periods (..) indicates the parent directory (the subdirectory level above this one).

▶ The WORD.DIR subdirectory has another subdirectory (DAVID.L) below it.

▶ WORD.DIR contains the file CHAP1.TXT, which is 23345 bytes, and was created or last modified at 11:13 AM on June 15, 1986.

3.3   Making a Subdirectory

You can create a subdirectory by entering a MKDIR (Make Directory) command from the keyboard or from a batch file. See the descriptions of the MKDIR and FOR commands in Section 6.4. You can shorten MKDIR to MD. In the MKDIR command, give the name of the directory you want to create.

MS-DOS creates a subdirectory as a special kind of file with a < DIR > attribute. The conventions for naming a directory are the same as for naming a file. You can use 1 to 8 alphanumeric characters, and you can include an optional filename extension of 1 to 3 alphanumeric characters preceded by a period.

For example, suppose you want to create a subdirectory named MGR1.NAM under the root directory on the diskette in the default drive. If you are in the root directory, you can enter

mkdir mgr1.nam↵

If you are in another subdirectory, you can enter

mkdir \mgr1.nam ↵

You can verify that MS-DOS created the subdirectory by entering the DIR command.

Once a directory exists, you can add other subdirectories to it. For example, to create a subdirectory called JOBS beneath MGR1.NAM, enter

md \mgr1.nam\jobs ↵

You can also create a subdirectory by first moving into the parent directory and then making the subdirectory. You move from one directory to another with the CHDIR command, described in Section 3.4. For example, enter these commands:

chdir mgr1.nam ↵
mkdir jobs ↵
dir ↵

MS-DOS moves to the MGR1.NAM subdirectory, creates the JOBS directory, and displays the current directory, MGR1.NAM:

```
Volume in drive A has no label
Directory of A:\mgrl.nam

.           <DIR>       6-30-86      9:00a
..          <DIR>       8-09-86     10:09a
JOBS        <DIR>       8-09-86     10:34a
        3 File(s)     370688 bytes free
```

If you create and save a file now, MS-DOS places it in the MGR1.NAM subdirectory. For example, if you enter this COPY CON command, MS-DOS creates and saves a file named TITLES in the MGR1.NAM subdirectory:

copy con titles ↵
Publication Manager ↵
Managing Editor ↵
Copy Editor ↵
^Z ↵

Now the name of the TITLES file, its size, and the date and time it was created are displayed in the MGR1.NAM directory:

```
Volume in drive A has no label
Directory of A:\mgr1.nam

.            <DIR>        6-30-86        9:00a
..           <DIR>        8-09-86       10:09a
JOBS         <DIR>        8-09-86       10:34a
TITLES          51 8-09-86             11:20a
        4 File(s)    370637 bytes free
```

The full path to JOBS is \MGR1.NAM\JOBS, and the path to the TITLES file is \MGR1.NAM\TITLES. Each \ indicates a level below the root directory. MGR1.NAM is the first subdirectory (indicated in the directory by the .), and JOBS is the subdirectory below it:

```
┌─────────────┐
│    Root     │
│  Directory  │
└──────┬──────┘
       │
┌──────┴──────┐
│  MGR1.NAM   │
│  Directory  │
└──────┬──────┘
       │
┌──────┴──────┐
│    JOBS     │
│  Directory  │
└─────────────┘
```
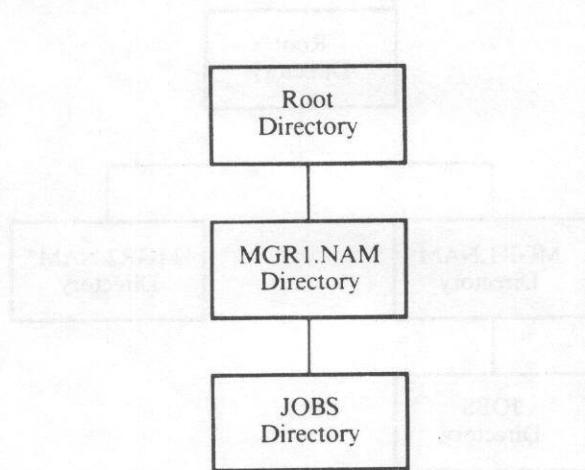
To add another subdirectory, such as MGR2.NAM, at the same level as MGR1.NAM, you must tell MS-DOS to start at the root directory:

mkdir \mgr2.nam ↵

The first \ in a MKDIR command indicates the root directory to MS-DOS. You can omit the initial \ if you are at the root directory.

MS-DOS creates MGR2.NAM and puts the new directory name in the root directory:
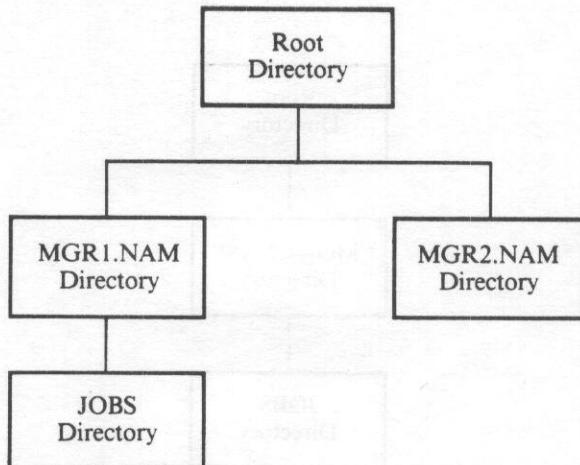
You can add a third directory level beneath JOBS, such as a subdirectory named BENEFITS. If you are currently at the JOBS subdirectory, you can create BENEFITS by entering

mkdir benefits⏎

If you are not at JOBS, enter

mkdir \mgr1.nam\jobs\benefits⏎

Now you have the following subdirectory system:

Root Directory

MGR1.NAM Directory

MGR2.NAM Directory

JOBS Directory

BENEFITS Directory
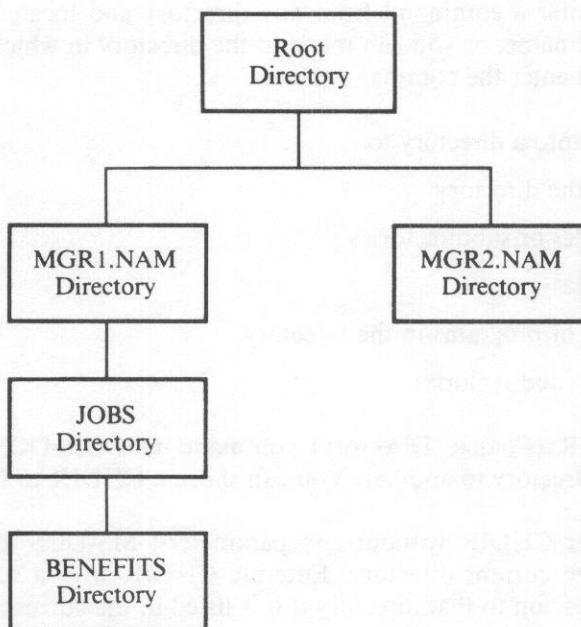
Because each subdirectory name is listed in the preceding directory, you cannot create a new subdirectory unless the parent directory exists. If you load MS-DOS and try to create several new subdirectories in the same command, such as:

mkdir mgr2.nam\mgr3.nam\mgr4.nam ↵

MS-DOS displays the message "Unable to create directory."

MS-DOS also does not allow you to give a file the same name as its subdirectory. Subdirectories are listed as files in a directory, and duplicate filenames are not allowed.

3.4 Moving into a Directory

Using a hierarchical file structure gives you two means of locating files. You can enter a command from any directory and locate the file by using a pathname, or you can move to the directory in which the file is located and enter the command there.

You move into a directory to:

▶ Display the directory

▶ Create files or subdirectories

▶ Delete files

▶ Use files or programs in the directory

▶ Remove subdirectories

The CHDIR (Change Directory) command tells MS-DOS to move from one directory to another. You can shorten CHDIR to CD.

If you enter CHDIR without any parameters, MS-DOS displays the name of the current directory. Entering CHDIR with a subdirectory name moves you to that directory if it is listed in the current directory.

If you enter CHDIR and a subdirectory name that is not listed in the current directory, MS-DOS displays the message "Invalid directory." This message also appears if you use CHDIR with a filename instead of a directory name.

Using the examples in Section 3.3, if you are in the root directory, you can move into the next subdirectory (MGR1.NAM) by entering

    chdir \mgr1.nam ↵

or

    cd \mgr1.nam ↵

The first \ in a CHDIR command indicates the root directory. If a directory name follows the backslash, it names the first subdirectory below the root directory. Each additional \ in the command is another subdirectory level in the path to the specified directory. The following command places you in the BENEFITS subdirectory:

chdir \mgr1.nam\jobs\benefits ↵

To move up to the MGR1.NAM subdirectory, enter

chdir \mgr1.nam ↵

To get to JOBS from MGR1.NAM (now the current directory), enter

chdir jobs ↵

If you are not at MGR1.NAM, you must indicate the path to JOBS:

chdir \mgr1.nam\jobs ↵

You can move to the level above the current directory level by typing

chdir .. ↵

This command moves you to the directory above the one you are currently using. For example, if you are at JOBS, MS-DOS moves up to MGR1.NAM.

Entering CHDIR with only a \ always returns you to the root directory. You should enter CHDIR \ when you finish working in a subdirectory. Otherwise you might save files in the wrong subdirectory.

3.5   Removing a Subdirectory

The RMDIR (Remove Directory) command removes an empty sub-directory. You can abbreviate RMDIR to RD.

Before you can remove a subdirectory, you must delete all filenames (with DEL) and all lower subdirectories (with RMDIR). An empty directory lists only the . and .. notations, which cannot be deleted because they are part of the MS-DOS directory system.

You can remove only one subdirectory at a time. For example, if you want to remove the BENEFITS subdirectory, first delete any files in BENEFITS. If you are in the BENEFITS subdirectory, use CHDIR to move to another directory. To remove the BENEFITS subdirectory while you are in the JOBS subdirectory, enter

   rmdir benefits ↵

You can also specify a full path from the root directory. For example, this command removes the BENEFITS subdirectory from the JOBS subdirectory if BENEFITS is already empty:

   rmdir \mgr1.nam\jobs\benefits ↵

If you are currently using the MGR1.NAM directory, you can specify the path to the empty BENEFITS subdirectory and remove it:

   rmdir jobs\benefits ↵

If the directory you name does not exist, or if you try to remove the current directory or one that is not empty, MS-DOS displays the message "Invalid path, not directory, or directory not empty."

4

Using the Line Editor (EDLIN)

EDLIN is a line editor program used mainly to create and edit source files (such as MS-DOS batch files). You can also use EDLIN to create or edit text. To input text such as memos or documents, however, it is easier and more efficient to use a word processing program.

You can use two types of commands for editing with EDLIN:

▶ File commands, which create or edit files

▶ Line commands, which edit characters within a single line of a file

The file commands are unique to EDLIN. The line commands are the MS-DOS command-line editing functions (described in Chapter 5).

To use EDLIN, load the program as shown in Section 4.1. Section 4.1.1 describes how to create a batch file; this batch file is used in examples throughout the chapter. Section 4.1.2 shows you how to edit an existing file.

Section 4.2 discusses how to use EDLIN file commands and introduces the format of the commands; Section 4.3 summarizes the file commands.

4.1 Invoking EDLIN

The EDLIN.COM file must be available on disk in order to load the program. If you are using subdirectories, MS-DOS must be able to find EDLIN.COM in the directory system. If EDLIN.COM is not in the current subdirectory, give its subdirectory location in the EDLIN command, or enter a PATH command to indicate its location to MS-DOS.

You load EDLIN and the file you want to edit at the same time. To invoke EDLIN, type

edlin filename ⏎

The filename can be either the name of an existing file or the name of the file you are creating. Include the drive name if the file is not on the default drive. To edit a data/text file in a subdirectory, either change to that subdirectory before loading EDLIN or give the subdirectory name with the filename when you invoke EDLIN. Chapter 3 describes how to use the MS-DOS subdirectory system.

4.1.1 Creating a File

To create a new file, enter EDLIN followed by the name of a file that does not already exist. Include the drive on which the file is to be saved unless you want the file to be saved on the default drive.

For example, to create the file NEWDISK.BAT in subdirectory MGR1.NAM on the diskette in drive B, enter

edlin b:\mgr1.nam\newdisk.bat ⏎

EDLIN creates a new, empty file, and displays the message "New file" and its command prompt (*):

```
New file
*_
```

To enter text into the file, type the Insert command (I) at the * prompt and press Enter. EDLIN then numbers and displays the first line for your entry:

```
New file
*I↵
     1:*_
```

Now you can type in the text of the file. End each line by pressing Enter. EDLIN displays the next line number.

Here is an example of an MS-DOS batch file created with EDLIN. This file displays the directory of a formatted diskette and checks the diskette directory for consistency.

```
A>edlin b:\mgrl.nam\newdisk.bat↵
New file
*I↵
     1:REM This is a file to check formatted
diskettes↵
     2:REM It is named NEWDISK.BAT↵
     3:PAUSE Insert formatted diskette in drive B↵
     4:DIR B:↵
     5:CHKDSK B:↵
```

You can stop text entry without saving any of the file by entering the Quit (Q) command. When you finish entering data, type a Ctrl-C (Interrupt) or Ctrl-Z (Terminate Text Entry) on the next line and press Enter. Either Ctrl-C or Ctrl-Z exits Insert mode and returns you to the * prompt.

To display the file you created, enter the List command (L). EDLIN displays the file and numbers the lines. (Line numbers are not saved in the file on disk.) When you insert new lines or delete existing lines, EDLIN adjusts the line numbers accordingly. The line numbers run from 1 through the number of the last line.

To save the file on disk, type the End command (E). You are returned to the MS-DOS command prompt.

4.1.2   Editing an Existing File

To edit an existing file, enter EDLIN, the drive and subdirectory containing the file, and the filename with its extension. For example, if you want to make changes to NOTICE.MEM in the WORD.DIR subdirectory on the default drive, enter

    edlin \word.dir\notice.mem ↵

EDLIN loads the file into memory. If EDLIN can load the entire file, it displays the message "End of input file" and the asterisk command prompt (*).

If EDLIN cannot load the entire file, it fills up ¾ of available memory with the first part of the file and does not display the "End of input file" message.

To edit a file larger than memory:

1. Display the portion of the file in memory with the List Line command (L).

2. Edit the portion of the file in memory.

3. Save the edited portion of the file on disk with the Write command (W). Add the rest of the lines from the file by using the Append command (A).

4.2   Using File Commands

Use the Insert command (I) to create an EDLIN file, as described in Section 4.1.1. After the file is created, you can use the EDLIN file commands to:

▶ Add lines from disk to memory if the file is too large for memory (Append).

▶ Delete one or more lines from the file (Delete).

▶ Modify existing lines (Edit).

▶ Add one or more lines of text to a file (Insert).

▶ Display the series of lines that you want to change (List).

▶ Replace previous text with new text within a certain number of lines in the file (Replace).

▶ Search through a specified number of lines in a file for a string of characters that you want to change (Search).

▶ Write edited portions of a large file from memory to disk (Write).

File commands are single letters (such as A for Append) with optional parameters. The parameters select, display, or change one or more lines in a file. You type the letter of the command and the parameters you want from the keyboard. The parameters and the commands are summarized in Table 4-1 and in Section 4.3.

4.2.1 File Command Parameters

The file command parameters are shown in Table 4-1.

Table 4-1: EDLIN File Commands and Parameters

COMMAND PARAMETER	PARAMETER DESCRIPTION	COMMANDS THAT USE PARAMETER
n	Number of lines.	
line	Represents the line number you select. For the line number, you can specify a decimal integer from 1–65534. If the number is larger than the lines in memory, the number indicates the line after the last existing line. Line numbers in a command must be separated from each other by a comma or a space.	A—Append lines W—Write lines
	Lines can also be specified by:	
	. (a period) represents the current line (marked on the screen by an * between the line number and the first character of the line).	D—Delete lines N—Edit lines I—Insert lines L—List text R—Replace a string of characters
	# (the pound sign) specifies the line after the last line (same as specifying the number larger than the last line number in the file).	
	↵ (Enter) directs EDLIN to use a default line number appropriate to the command.	
?	Directs EDLIN to query for a yes or no response to an Ok? prompt.	R—Replace a string of characters S—Search for a string
string	Represents one or more characters to be found or replaced. Each string must end with a Ctrl-Z or Enter. There should be no spaces between the string parameters unless the space is part of the string. Do not enter spaces between string parameters and the command.	R—Replace a string of characters S—Search for a string in a line

4.2.2 File Command Conventions

Some conventions are common to all the file commands. With any file command you can:

▶ Type commands with or without a space between the line number and the command. For example, enter

5D

or

5 D

▶ Reference lines relative to the current line. On the screen, the current line has an * between the line number and the first character of the line. Include a plus sign with a line number in a command to indicate the number of lines following the current line. A minus sign with a line number in a command indicates the number of lines before the current line.

For example, this command directs EDLIN to begin inserting text one line before the current line on the screen:

-1I

▶ Issue multiple commands on one command line. Separate multiple commands in a line with a semicolon. For example, if you want to edit a line and then display other lines, enter

10;-2, + 1L⏎

This command lets you edit line 10, then display (List) the two lines before line 10 and the line following line 10. The comma after − 2 is a syntax element for **line,line**—see the List command in Section 4.3.

If you are issuing a multiple command with a string parameter, as in a Search or Replace command, separate the commands with a Ctrl-Z. For example, this command

2,10 SThis string^Z-2; + 2L⏎

searches for the first occurrence of the characters "This string" in lines 2 through 12. It displays the 2 lines before the line that contains "This string" and the 2 lines after it.

If "This string" cannot be found, the 2 lines before the current line (noted on the screen with the *) and the 2 lines after the current line are displayed.

4.3 Descriptions of File Commands

This section presents detailed descriptions of each EDLIN command for editing files. The descriptions are listed alphabetically.

Append Lines (A)

[n]A↵

A adds the number of lines, specified by **n**, from disk to the end of the lines currently in memory. **n** represents the number of lines.

Use this command for large files that cannot fit into memory at one time. Edit the lines in memory, write them to disk with the Write command (W), then use the Append command (A) to put the rest of the file in memory.

If you enter a number with the A command, that number of lines is appended to the part of the file currently in memory. If no number is entered, the lines are read into memory until memory is ¾ full. If memory is already full, no more lines can be appended.

When the last line of the file is in memory, EDLIN displays

```
End of input file
*_
```

Delete Lines (D)

[line][,line]D ↵

D deletes the lines specified. **line** can be any of the parameters for a line described in Table 4-1. Without parameters, D deletes the current line.

line
> deletes the specified line.

,line or **,#**
> deletes the current line (marked by the *), the specified line, and all lines between the current line and the specified line.

line, or **line,line** or **line,#**
> deletes the specified lines and all lines between them.

The line immediately after the deleted lines becomes the current line (marked by the *).

For example, suppose you have the file NEWDISK.BAT:

```
1:REM This is a file to check formatted diskettes
2:REM This is a batch file created by EDLIN
3:REM This file is named NEWDISK.BAT
4:PAUSE Insert formatted diskette in drive B
5:DIR B:
6:CHKDSK B:
```

To delete lines 2 and 3, enter

2,3 D ↵

To verify the result, enter the List command (L):

```
*L↵

     1:REM This is a file to check formatted
 diskettes
     2:*PAUSE Insert formatted diskette in drive B
     3:DIR B:
     4:CHKDSK B:
```

After lines 2 and 3 are deleted, line 4 becomes line 2 and is marked by
the asterisk as the current line.

To delete one line, such as line 3, enter

   3D↵

To see the result, specify the List command (L):

```
*L↵

     1:REM This is a file to check formatted
 diskettes
     2:PAUSE Insert formatted diskette in drive B
     3:*CHKDSK B:
```

When line 3 is deleted, line 4 becomes line 3 and is marked as the
current line.

To delete the current line, now marked as line 3, enter

   D↵

Specify the L command at the * prompt:

```
1:REM This is a file to check formatted diskettes
2:PAUSE Insert formatted diskette in drive B
```

Edit Line

[n]↵

[.]↵

[#]↵

n represents the line number.

. indicates the current line.

# indicates the line after the last line in the file.

See Table 4-1 for an explanation of the line number parameters.

EDLIN displays the line number and the text on that line. On the next line, it repeats the line number and displays the asterisk prompt and the cursor. You can start editing the line at the cursor position. You can use the command-line editing keys described in Section 5.2 to edit the line.

If you do not want to change the current line (marked by the *), and the cursor is at the beginning of the line, press the Enter key to accept the line as it is. If you press Enter while the cursor is in the middle of the line, the rest of the line is deleted.

For example, you can select line 1 with the Edit Line command:

1 ↵

EDLIN displays the specified line:

```
1:*REM This is a file to check formatted diskettes
1:*_
```

You can now type in a new line 1, or you can use the line-editing commands to edit the current line (the template, described in Section 5.2).

End Edit (E)

E ↵

E ends the editing session, saves the edited file on disk, changes the original file's extension to .BAK, and exits to MS-DOS. If the file was newly created during the editing session, no .BAK file is created.

Because the E command takes no parameters, you cannot select the drive on which to save the file. If you did not designate the drive when you invoked EDLIN, the file is saved on the default drive.

If the diskette does not have enough free space for the entire file to be written, the write is aborted and the edited file is lost. Part of the file, however, may be written to the diskette.

Insert Lines of Text (I)

[line]I ↵

I inserts line(s) of text immediately before the specified line.

line can be any of the parameters for a line described in Table 4-1.

You must enter the I command to create a new file. The first insert begins with line number 1. EDLIN numbers each new line sequentially each time you press Enter. EDLIN remains in insert mode until you enter a Ctrl-Z or a Ctrl-C as your last line, ended with Enter.

If you insert lines between existing lines in a file, all line numbers following the inserted section are incremented by the number of lines inserted.

If you do not include a line number, the new lines are inserted immediately before the current line. If the specified line number is larger than the last line number or if # is specified as the line number, the inserted lines are appended to the end of the file. In this case, the last line inserted becomes the current line.

For example, suppose you have the following file:

```
1:REM This is a file to check formatted diskettes
2:PAUSE Insert formatted diskette in drive B
3:DIR B:
4:CHKDSK B:
```

To add lines before line 2, enter

2 I ↵

EDLIN displays

```
2:*_
```

Type

REM This file is a batch file.↵
REM This batch file is called NEWDISK.↵
^Z↵

You can display the new file with the L command:

```
1:REM This is a file to check formatted diskettes
2:REM This file is a batch file
3:REM This batch file is called NEWDISK
4:*PAUSE Insert formatted diskette in drive B
5:DIR B:
6:CHKDSK B:
```

To insert lines immediately before the current line (line 4), enter

I↵

The screen displays

```
4:*_
```

Enter

REM NEWDISK was created with EDLIN.↵
^Z↵

Type the L command to list the file:

```
1:REM This is a file to check formatted diskettes
2:REM This file is a batch file
3:REM This batch file is called NEWDISK
4:REM NEWDISK was created with EDLIN
5:*PAUSE Insert formatted diskette in drive B
6:DIR B:
7:CHKDSK B:
```

List Lines (L)

[line][,line]L ⏎

line can be any of the parameters for a line described in Table 4-1.

L lists the specified range of lines, as follows:

▶ With no parameters, L lists the 23 lines centered around the current line (marked by an *). The 23 lines are the 11 lines preceding the current line, the current line, and the 11 lines following the current line.

▶ **line,line** or **line,#** lists the specified range of lines.

▶ **,line** lists the range of lines preceding the specified line.

▶ **line,** lists a range of lines following the specified line.

Assume the following file exists and is ready to edit:

```
1:REM This is a file to check formatted diskettes
2:REM It is named NEWDISK.BAT
3:PAUSE Insert formatted diskette in drive B
4:DIR B:
5:CHKDSK B:
```

To list the entire file, enter

L↵

All five lines are displayed, because L without parameters displays up to 23 lines of text.

To list a range of lines preceding line 3, enter

,3L↵

EDLIN displays

```
1:REM This is a file to check formatted diskettes
2:REM It is named NEWDISK.BAT
3:PAUSE Insert formatted diskette in drive B
```

To list a range of lines following line 3, enter

3,L↵

EDLIN displays

```
3:PAUSE Insert formatted diskette in drive B
4:DIR B:
5:CHKDSK B:
```

Quit Edit (Q)

Q⏎

Q quits the editing session without saving the editing changes, and exits to MS-DOS. The Q command is a quick way to exit an editing session. When you enter the Q command, EDLIN displays the following message:

```
Abort edit (Y/N)?
```

Press Y to quit the editing session; press N (or any other key except Ctrl-C) to continue the editing session.

Replace String (R)

[line][,line][?]Rstring1[^Z]string2⏎

R replaces string1 with string2, in the range of lines specified by **line**. **line** can be any of the parameters for a line described in Table 4-1. If no line is indicated, R operates on the line after the current line (marked by an *) through the end of the file.

▶ **line** is the specified line to the end of the file.

▶ **line,line** is the first specified line through the second specified line.

▶ **,line** is the line after the current line through the specified line.

▶ **.,line** is the current line through the specified line.

▶ **line,.** is the range of lines from the specified line through the current line.

? prompts you to accept each string replacement or deletion. When EDLIN encounters a matching string of text, it displays the line and the prompt "O.K.?" If you press Y or Enter, EDLIN makes the replacement or deletion and searches for the next matching string. If you press any key other than Y or Enter, EDLIN leaves the string as is and searches for the next matching string.

string1 is the string of characters to be replaced. **string2** is the string of characters to replace string1. Ctrl-Z separates string1 from string2.

If you specify only one string of characters, R deletes all occurrences of the string within the specified range of lines.

Suppose the following file exists and is ready for editing:

```
1:REM This is a file to check a formatted diskette
2:REM It is named NEWDISK.BAT
3:PAUSE Insert a formatted diskette in drive B
4:DIR B:
5:CHKDSK B:
```

To replace all occurrences of "a" with "one" in lines 1 through 3, enter

1,3Ra^Zone ↵

EDLIN displays

```
      1:*REM This is one file to check one formonetted
diskette
      2:It is nonemed NEWDISK.BAT
      3:PAUSE Insert one formonetted diskette in
drive B
```

Some of these replacements produce nonsense words. To avoid these unwanted substitutions, add the ? parameter:

1,3?Ra˄Zone↲

Now each specified line is displayed, and you can confirm the changes. For example:

```
      1:*REM This is one file to check a formatted
diskette
O.K.?y
      1̄:*REM This is one file to check one
formatted diskette
O.K.?y
      1̄:*REM This is one file to check one
formonetted diskette
O.K.?n
2:REM It is nonemed NEWDISK.BAT
O.K.?n
3:PAUSE Insert one formatted diskette in drive B
O.K.?y
```

After making the replacements you want, enter an L command to list the edited file:

```
*L↲

    1:REM This is one file to check one formatted
diskette
    2:REM It is named NEWDISK.BAT
    3:*PAUSE Insert one formatted diskette in
drive B
    4:DIR B:
    5:CHKDSK B:
```

Search for a String (S)

[line][,line][?]Sstring ⏎

S searches for the string in the range of lines specified by **line** and displays the first line containing the string. This line becomes the current line if the ? parameter is not specified.

line can be any of the parameters for a line described in Table 4-1. The syntax for **line** is given in the Replace command description. If you do not specify a line, S operates on the line after the current line through the end of the file.

? prompts you to accept the displayed line as the current line. When EDLIN encounters a matching string of text, it displays the line and the prompt "O.K.?" If you press Y or Enter, the displayed line becomes the current line. If you press any key other than Y or Enter, EDLIN searches for the next matching string.

Assume the following file exists and is ready for editing:

```
1:REM This is a file to check a formatted diskette
2:REM It is named NEWDISK.BAT
3:PAUSE Insert a formatted diskette in drive B
4:DIR B:
5:CHKDSK B:
```

To search for the first "in" within lines 1 through 3, enter

1,3Sin ⏎

EDLIN displays

```
3:PAUSE Insert a formatted diskette in drive B
```

To find "DIR" past line 3, specify

3SDIR ↵

The search starts at line 3 and continues to the end of the file. EDLIN displays the line containing the first "DIR":

```
4:DIR B:
```

To search the file for the first occurrence of "B", at the * prompt enter

?SB ↵

EDLIN displays

```
     2:REM It is named NEWDISK.BAT
O.K.?_
```

If you respond Y, the search ends and you return to the * prompt. To continue the search, respond N.

Write Lines (W)

[n]W ↵

W writes lines from memory to disk and makes room in memory for more lines to be appended (with the Append command) from disk to memory.

n represents the number of lines to be written.

If no parameter is given, lines are written to disk until memory is only ¼ full. If a number is given, the specified number of lines is written to disk. As lines are written, subsequent lines are renumbered beginning with 1.

An A command following the W command appends lines from disk to any line remaining in memory.

5

MS-DOS Command Editing, Filtering, and Piping

This chapter describes how to enhance your control of MS-DOS commands. Section 5.1 describes the control character editing functions you use during input. Section 5.2 describes the special MS-DOS editing keys and how to use function keys to redisplay and edit parts of the previous command line (the template).

Sections 5.3 and 5.4 demonstrate two major processes—filtering and piping data. These processes enable you to reorganize and output data between files or devices, and to use the output from one command as input to another command in the same command line.

5.1 Control Character Functions

A control character function is a function that affects the command line. To type a control character, such as Ctrl-Break, you hold down the Ctrl (Control) key while you press the specified key. Table 5-1 lists the MS-DOS control character functions.

Table 5-1: Control Character Functions

CONTROL CHARACTER	FUNCTION
Ctrl-PrtSc	Toggles to echo screen output to the line printer.
Ctrl-Break	Aborts current command.
Backspace	Removes the last character from command line, and erases the character from the screen.
Ctrl-Enter	Inserts a line feed character (physical end-of-line) but does not empty the command line. Use Ctrl-Enter to extend the current logical line beyond the physical limits of the screen.
Shift-PrtSc	Dumps the current contents of the screen (25 lines) to the printer.
Ctrl-Num Lock	Suspends output display on the screen. Press any key to resume. Same as Ctrl-S.
Esc	Cancels the current line and outputs a backslash (\), carriage return, and line feed, thus clearing the working command line. The special editing command template is not affected.

Home of Victor Computers

5.2 Special MS-DOS Editing Keys

The way MS-DOS uses special editing keys differs from the way in which most operating systems handle command input. You do not have to type the same sequences of keys repeatedly because the last command line is automatically placed in a special storage area called a **template**. By using the template and the special editing keys, you can take advantage of the following MS-DOS features:

▶ You can repeat a command line instantly by pressing two keys.

▶ If you make a mistake in the command line, you can edit it and retry the command without retyping the entire command line.

▶ By pressing a special editing key, you can edit and execute a command line that is similar to a preceding command line with a minimum of typing.

Figure 5-1 shows the relationship between the command line and the template.
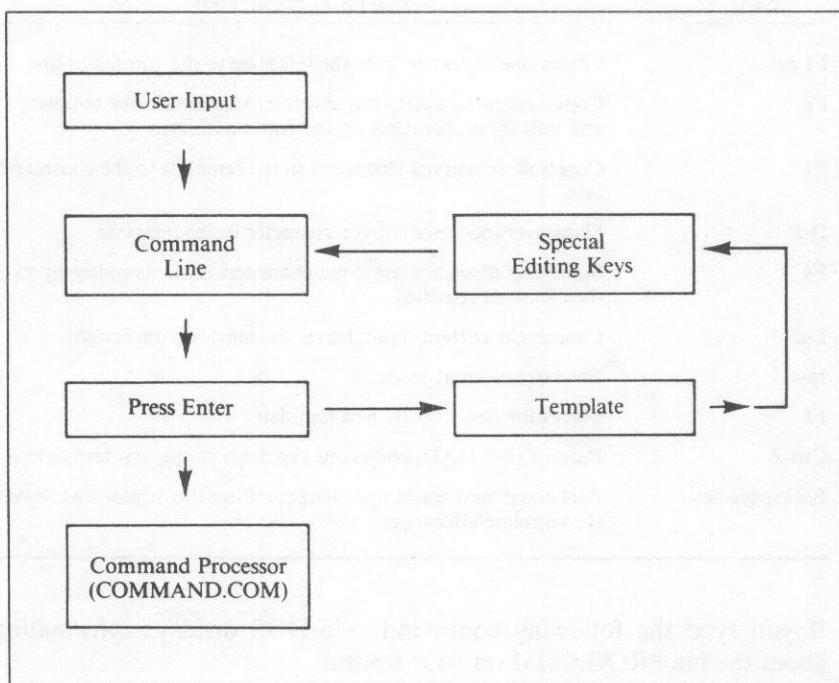
Figure 5-1: The Command Line and the Template

As shown in the figure, you type a command to MS-DOS on the command line. When you press the Enter key, the command is automatically sent to the command processor (COMMAND.COM) for execution. At the same time, a copy of this command is sent to the template. Once a copy of the command is in the template, you can recall the command or modify it with the MS-DOS special editing keys. Table 5-2 lists these keys.

Table 5-2: Special Editing Functions

KEY	EDITING FUNCTION
F1 or →	Copies one character from the template to the command line.
F2	Copies characters up to the character specified 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 character specified.
Esc	Cancels the current input; leaves the template unchanged.
Ins	Enters/exits insert mode.
F5	Makes the new line the new template.
Ctrl-Z	Puts a Ctrl-Z (1AH) end-of-file character in the new template.
Backspace or ←	Backspaces and erases one character from the display line; leaves the template unchanged.

If you type the following command, MS-DOS displays information about the file PROG.COM on your screen:

dir prog.com ↵

The command line is also saved in the template. To repeat the command, just press F3; the command is displayed on the screen. Then press Enter. Pressing the F3 key causes the contents of the template to be copied to the command line; pressing Enter causes the command line to be sent to the command processor for execution.

If you want to display information about a file named PROG.ASM, you can use the contents of the template and type

(F2)c ↵

where (F2) represents function key F2. Pressing F2 and then C copies all characters from the template to the command line, up to but not including C. MS-DOS displays

```
DIR PROG._
```

In the example above, the underline represents the cursor. Now type

asm

MS-DOS displays

```
DIR PROG.ASM_
```

The command line DIR PROG.ASM is now in the template and ready to be sent to the command processor for execution. To do this, press Enter.

Now assume that you want to execute the following command:

type prog.asm

To do this, type

type(Ins)(sp)(F3)⏎

where (Ins) represents the Insert key, (F3) is function key F3, and (sp) is the Spacebar. As you type, the characters are entered directly into the command line and overwrite corresponding characters in the template. Thus, in the preceding example, the characters TYPE and PROG.ASM replace the characters DIR in the template. To insert a space between TYPE and PROG.ASM, press Ins and then the Spacebar. Finally, to copy the rest of the template to the command line,

press F3 and then Enter. The command TYPE PROG.ASM is pro-
cessed by MS-DOS, and the template becomes TYPE PROG.ASM.

If you misspell TYPE as BYTE, you will receive an error message. But
you can save the misspelled line before you press Enter by creating a
new template with the F5 key:

byte prog.asm(F5)

You can now edit this erroneous command by typing

t(F1)p(F3)

The F1 key (or the → key) copies a single character (in this case, Y)
from the template to the command line. Enter P, then press F3 to copy
the rest of the template to the command line. The correct command
line is displayed:

```
type prog.asm
```

Or you can use the same template containing BYTE PROG.ASM and
then use the Del and Ins keys to achieve the same result:

(Del)(Del)(F1)(Ins)yp(F3)

To illustrate how the command line is affected as you type, examine
the keys typed on the left; their effect on the command line is shown
on the right:

KEY	DISPLAY	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
(F3)	TYPE PROG.ASM	Copies rest of template

The Del key does not change the command line; it affects the template by deleting the first character. Similarly, F4 deletes characters in the template, up to but not including a given character.

5.3 Filtering Data

MS-DOS can sort through the data in a text file and re-arrange it. This process is called **filtering**. These commands filter data:

▶ FIND searches for a constant string of alphanumeric characters.

▶ SORT rearranges the data in the order that you specify in the command. Order is predefined by the ASCII collating sequence. See Appendix A for an ASCII conversion table.

▶ MORE displays one screen of data at a time; press Enter to display another screen of data. See Section 6.4 for information on the MORE command.

You can use these filtering commands with any MS-DOS files, such as VBASICA files. The examples in this section use the following data file, PHNLST.DAT, containing four records of names and phone numbers:

```
Washington, George 301 328-4370
Kennedy, John 617 736-2867
Lee, Robert 703 532-3370
Lincoln, Abraham 309 473-2800
```

You can filter the data by area code with the FIND and SORT commands. In the following example, FIND searches for a numeric string of characters, the 309 area code:

```
find "309" b:phnlst.dat ↵
```

MS-DOS displays

```
----------b:phnlst.dat
Lincoln, Abraham 309 473-2800
```

The SORT command can arrange data alphabetically or the reverse, Z through A. The following command tells MS-DOS to arrange the PHNLST.DAT data records alphabetically and to output the sorted data to the screen:

sort < b:phnlst.dat > con⏎

The < and > symbols redirect input and output. The < in the previous command tells MS-DOS that the file PHNLST.DAT on drive B is the input for the SORT command. The > symbol tells MS-DOS to send the output (the sorted data) to the display unit (CON).

The preceding command displays this sorted data:

```
Kennedy, John   617 736-2867
Lee, Robert 703 532-3370
Lincoln, Abraham 309 473-2800
Washington, George 301 328-4370
```

You can also tell MS-DOS to create a new file and send sorted data to the new file. For example, if you name the file ALPHA.NUM in place of CON in the previous example, MS-DOS sends the alphabetized file records to ALPHA.NUM on drive B:

sort < b:phnlst.dat > b:alpha.num⏎

5.4 Piping Data

To **pipe** data means to enter more than one command on a command line, separating commands with a vertical bar (|), the pipe separator. By piping commands you can use the output from one command as the input for another command. You can combine piping with filtering to re-arrange existing files into new sorted files.

You can also perform these tasks:

▶ Filter data from an existing data file and pipe it to a new file or other output source, such as the screen (CON) or printer (PRN).

▶ Filter columnar data, such as a directory, by column number (the screen column where file size, date, or time is listed).

▶ Use batch processing to make filtering and piping data even easier.

5.4.1 Filtering and Piping Data Files

Using the sample file shown in Section 5.3, PHNLST.DAT, you can filter the data by area code. You can then pipe the filtered area codes to the AREA.COD file by entering

 find "301" b:phnlst.dat | sort > b:area.cod ↵

The | pipe symbol tells MS-DOS to take the output from the command on the left side of the bar and use it as input for the command on the right side of the bar. The > symbol in the command sends all the records containing the 301 area code to a new file named AREA.COD.

The following command uses the >> symbol to append all the records containing the 309 area code to the end of the AREA.COD file:

 find "309" b:phnlst.dat | sort >> b:area.cod ↵

To verify that the sorted records are in AREA.COD, specify

type b:area.cod ↵

Enter the following command:

type b:phnlst.dat I sort > prn ↵

MS-DOS gets the output of the TYPE B:PHNLST.DAT command and uses it as input for the SORT command. This SORT command sorts the file alphabetically and sends the sorted data to the printer.

5.4.2 Filtering and Piping Columnar Data

Using the SORT command, you can pipe and sort columns of data. For example, type

dir I sort / + 16 I more ↵

MS-DOS pipes the output of the DIR command as input to the SORT filter command. The sorting starts in column 16 of the directory display (file size). MS-DOS sends the directory, sorted by file size, to the display unit. MORE displays the directory one screen at a time with the message --MORE-- at the bottom of the screen. You can display the next screenful of directory information by pressing Enter.

5.4.3   Filtering and Piping Batch Files

You can use filtering and piping in batch files. The next example creates a batch file PHONES.BAT that searches for a value (the %1 parameter) in the PHNLST.DAT file, sorts the records containing that value, and displays those records onscreen:

```
copy con phones.bat↵
find "%1" phnlst.dat I sort > ↵
^Z↵
```

You can use PHONES.BAT to search for any area code you want. Type PHONES and the area code you want to have sorted from the PHNLST.DAT file, such as:

```
phones "301"↵
```

All the records from PHNLST.DAT with area code 301 are displayed.

6

MS-DOS Commands

This chapter presents an overview of MS-DOS commands. Section 6.1 lists and describes the two types of MS-DOS commands (internal and external) as well as the batch commands (internal commands used in batch files). Section 6.2 describes the notation used in the syntax of commands in this manual.

Section 6.3 describes the special commands you use when you alter the system configuration contained in the CONFIG.SYS file. Finally, Section 6.4 presents the syntax of all the MS-DOS commands (excluding EDLIN and the system configuration commands). Commands are listed alphabetically and examples are included for most commands.

6.1   Overview of Commands

You use the MS-DOS commands to set up and manage your files and directories, the diskettes containing your files, and the devices attached to your computer. This overview describes some of the MS-DOS commands and what you can do with them. For a complete listing of the commands, see Section 6.4.

You use the MS-DOS file commands to:

▶ Copy, delete, rename, display, or verify files (COPY, DEL, REN, TYPE, and VERIFY)

▶ Copy the operating system files from one drive to another (SYS)

▶ Change the file format or rearrange file contents (SORT and FIND)

▶ Queue a list of files for printing (PRINT)

▶ Create batch files and control their processing (COPY CON, ECHO, SHIFT, and so on)

You use the MS-DOS directory commands to:

▶ Display the directory of a diskette (DIR)

▶ Create or remove directories (MKDIR and RMDIR)

▶ Change from one subdirectory to another, or check your position in the directory hierarchy (CHDIR)

You use the MS-DOS disk commands to:

▶ Prepare blank diskettes for MS-DOS use (FORMAT)

▶ Copy the contents of one diskette to another (DISKCOPY)

▶ Check the amount of space on a diskette, and check a directory for errors in file storage (CHKDSK)

▶ Recover data from a bad sector on a disk (RECOVER)

The MS-DOS device commands (CTTY) control the screen, printer, and disk drives. These commands display information about your diskettes and disk drives. They can also change devices, such as changing output from the screen to the printer.

The system configuration commands (such as BREAK, BUFFERS, and DEVICE) enable you to control parts of the operating environment. All these commands are used in the CONFIG.SYS file, which sets up the environment when MS-DOS loads into memory. The BREAK command can also be used interactively. See Section 6.3 for a description of the configuration commands.

The MetaWare commands (CONCAT, FGREP, LS, MV, SEARCH, TAIL, UNIQ, and WC) are UNIX™-like extensions that offer shortcuts for working with your files. You can use these utilities to search through files or directories for specified text and compare, merge, manipulate, list, or move files.

Other MS-DOS commands affect the way the operating system works. For example, there are commands to set up constants in the operating environment, and commands to change the system command prompt.

6.1.1   Internal and External Commands

There are two types of MS-DOS commands: internal and external. Internal commands are a set of commands that exist within MS-DOS. External commands are stored on disk as separate files. External commands have either a .COM or .EXE file extension. External commands are also known as utility programs.

The command processor program interprets and processes all the MS-DOS commands, both internal and external. The command processor is in the system file COMMAND.COM on the MS-DOS system diskette. When you load MS-DOS, COMMAND.COM (or any command processor file named in the CONFIG.SYS file) is automatically loaded into memory.

The internal commands are part of the COMMAND.COM file. External commands, on the other hand, must be individually loaded into memory by COMMAND.COM each time you use that command. The internal commands remain in memory as long as COMMAND.COM is in memory.

The MS-DOS command interface is illustrated in Figure 6-1.

Other MS-DOS commands affect the way the system redirects work.
For example, the ... commands direct output and input to or of the
environment, and ... commands redirect the system's console output.


```
Command                                    Internal
Line      ──▶   COMMAND.COM                 Commands


                 │                    ▲
                 ▼                    │

      Memory                     Disk Space
      (External commands         ▶ External commands
      loaded and executed by       (.COM and .EXE files)
      COMMAND.COM)               ▶ Batch files (.BAT files)
                                 ▶ Data files
```

Figure 6-1: MS-DOS Command Interface

Internal Commands

The internal commands are:

BREAK	DEL	PATH	TIME
CHDIR	DIR	PROMPT	TYPE
CLS	EXIT	REN	VER
COPY	HISTORY	RMDIR	VERIFY
CTTY	MKDIR	SET	VOL
DATE			

These internal commands are not displayed in the directory. When
you enter an internal command, it executes immediately whenever
COMMAND.COM is in memory.

Some internal commands can use paths to obtain files from the MS-DOS subdirectory system (see Chapter 3). These commands are DEL, DIR, COPY, CHDIR, MKDIR, TYPE, and PATH.

External Commands

The external commands are:

ASSIGN	EXE2BIN	KEYBUK	SELECT
ATTRIB	FGREP	LABEL	SHARE
BACKUP	FIND	LS	SORT
CHKDSK	FORMAT	MODE	SUBST
COMMAND	GRAFTABL	MORE	SYS
COMP	GRAPHICS	MV	TAIL
CONCAT	JOIN	PRINT	TREE
DEBUG	KEYBFR	RECOVER	UNIQ
DISKCOMP	KEYBGR	RESTORE	WC
DISKCOPY	KEYBSV	SEARCH	
EDLIN			

Section 6.4 describes all the external commands except EDLIN (see Chapter 4).

Like all programs, the external commands are stored in command files on disk. You can identify them by the extension .COM or .EXE. You should keep all the MS-DOS commands in one directory.

Before you can successfully issue an external command, MS-DOS has to know where to locate the command file. Use either the CHDIR command to change to the directory where you keep your command files, or use the PATH command to specify a path to that directory. When you enter the command, MS-DOS reads the command from that directory. If MS-DOS cannot find the command file, the message "Bad command or filename" is displayed.

6.1.2   Batch Commands

Batch commands are used only in batch files (see Section 2.5). Because batch commands are internal commands, COMMAND.COM must be loaded in main memory whenever you run a batch file that uses any of these commands:

ECHO   PAUSE
FOR–IN–DO   REM
GOTO   SHIFT
IF–EXIST

6.2   Command Syntax and Notation

Descriptions of all the VICTOR MS-DOS commands are listed alphabetically in Section 6.4. In each description, the syntax or format for each command is given first. Then the elements of the command are defined, and examples are presented to show how to use the commands. In examples, the entire command line is lowercase except for elements that require uppercase, such as proper nouns or command switches that are not valid in lowercase.

The notation used for command syntax is as follows.

COMMAND NAMES
are shown in uppercase. You can, however, enter command names in uppercase or lowercase.

LITERAL PARAMETERS
are shown in uppercase. These are parts of a command that must be entered literally if you want to include them. You can enter the parameters in uppercase or lowercase, except as noted in the command descriptions.

variable parameters
are shown in lowercase. You can enter command parameters in uppercase or lowercase.

[ ]

enclose optional parameters.

...

is an ellipsis indicating that you can repeat the preceding parameter.

filename.ext, d:

represent the type of entry you must make. For example, when you see "filename.ext", you must enter the name of a specific file (with or without wildcard characters).

path

is the full or relative path through the directory hierarchy to the sub-directory or file named in the command. In this chapter, the syntax for paths is:

[d:][[\ ]directoryname[\ directoryname...]]

If the file or subdirectory you want is in the current directory on the default drive, you can omit the directory and drive names. The first \ in a path always refers to the root directory. Each additional \directoryname refers to a lower level subdirectory. Path syntax is fully described in Chapter 3.

/

indicates that the parameter (such as /C) is a switch. A switch causes MS-DOS to process the command as altered by the specified switch. When you enter a switch in the command line, separate it from commands and other switches with a space.

|

indicates that data is to be piped (see Chapter 5). The output generated by the command on the left side of the bar is sent as input to the command on the right side of the bar. Separate the | from other command elements with a space.

<

indicates that the input to the preceding command is to come from a source other than the keyboard. Enter a space before and after the <.

>

indicates that the output from the preceding command is to be sent to a file or device (such as the printer). Enter a space before and after the > .

>>

indicates that the output from the command is to be sent to the end of the file named in the command. Separate the >> from other command elements with a space, but do not type a space between the > s.

↵

indicates the Enter key.

(sp)

indicates the Spacebar.

^

indicates the Ctrl (Control) key. A Ctrl-key sequence (pressing Ctrl and another key) is shown as Ctrl- and the letter or number, such as Ctrl-C.

You must include all punctuation (commas, colons, question marks), the symbols \, ǀ, <, >, and >>, and spaces where shown, except for [ ], which indicate optional parameters, and the ellipsis. The ellipsis indicates that a parameter can be repeated.

Note: Because the symbols \, ǀ, <, and > are literal parts of MS-DOS commands, they cannot be used in a filename. If you have any existing files that have \, ǀ, <, or > in their filenames, you must rename the files before you can use them with MS-DOS Version 3.1 or later. Use a previous version of MS-DOS to rename the files.

6.3   System Configuration Commands

In many cases, there are installation-specific settings for MS-DOS that need to be configured at system startup. An example of this is a standard device driver, such as an on-line printer.

The MS-DOS configuration file CONFIG.SYS allows you to configure your system. With this file, you can add device drivers to your system at startup. The configuration file is an ASCII file that has certain commands for MS-DOS startup (boot). This section describes the commands you can use in CONFIG.SYS. You do not need to change this file unless you want to change the operating environment to fit your needs.

The boot process is as follows:

1. The disk boot sector is read. The boot sector contains enough code to read MS-DOS code and the installation's BIOS (machine-dependent code).

2. The MS-DOS code and BIOS are read.

3. A variety of BIOS initializations are done.

4. A system initialization routine reads CONFIG.SYS, if it exists, to perform device installation and other user options. Then it processes the command interpreter, finishing the MS-DOS boot process.

When MS-DOS loads into memory, it checks the root directory for the system configuration file CONFIG.SYS. This file contains system initialization routines that tell MS-DOS:

▶ When to allow Ctrl-C or Ctrl-Break to interrupt operations (BREAK)

▶ The number of buffers to allocate for input and output data (BUFFERS)

▶ The standards associated with particular countries (COUNTRY)

▶ The device drivers to be installed (DEVICE)

▶ The number of files that can be opened by file control blocks (FCBS)

▶ The number of files that can be open at the same time (FILES)

▶ The maximum number of drives you can access (LASTDRIVE)

▶ The name of the command processor to be loaded (SHELL)

To change the CONFIG.SYS file, use any text editor that operates under MS-DOS, such as EDLIN. You can also create a new CONFIG.SYS file with the COPY CON command. The changes you make to the file become effective the next time you load MS-DOS.

The rest of this section describes the system configuration commands that can appear in CONFIG.SYS.

BREAK

BREAK = ON↵

   or

BREAK = OFF↵

The default, BREAK = OFF, tells MS-DOS to check for a Ctrl-C interrupt only during screen, keyboard, printer, or asynchronous communication adapter operations. BREAK = ON tells MS-DOS to check for a Ctrl-C interrupt during all operations, including whenever MS-DOS is processing a program. You can change the setting by issuing the interactive BREAK command, described in Section 6.4. If you set BREAK = ON, you can stop programs by pressing Ctrl-C or Ctrl-Break.

BUFFERS

BUFFERS = xx ⏎

where **xx** is a number between 1 and 99. If you don't run a
CONFIG.SYS file with the BUFFERS command, the default value is
2. A buffer is the amount of memory allocated for the reading of data
from disk and the writing of data to disk. If your application does
sequential access to the data on disk, you can use a small number of
buffers. If you are using a database with random access, 10 to 20
buffers is adequate. A larger number of buffers speeds disk access. If
the number gets too large, however, you use more memory and slow
operations.

COUNTRY

COUNTRY = nnn ⏎

where **nnn** is a number between 001 and 099. The COUNTRY com-
mand selects configuration standards according to the given country
code. Some of the standards affected are the currency separator and the
format for the DATE command. The numbers you can use with
COUNTRY are the country's international telephone code, such as:

001	US	044	United Kingdom
032	Belgium	046	Sweden
033	France	047	Norway
034	Spain	049	Germany
039	Italy	061	Australia
041	Switzerland	081	Japan

MS-DOS uses the COUNTRY = nnn parameter in the CONFIG.SYS
file to set the TIME and DATE format.

DEVICE

DEVICE = [path]filename.ext[parameters] ↵

This command installs the device driver from the specified file. The default is the device drivers for the standard screen, keyboard, printer, auxiliary device, floppy disk drive, and fixed disk. DEVICE also lets you include device drivers that you have written or those provided with MS-DOS 3.1, such as ANSI.SYS and VDISK.SYS. For each device driver, you must include a DEVICE command in the CONFIG.SYS file.

path
is the path to the device driver file, including drive and subdirectory names where necessary.

filename.ext
is the name of the file containing the device driver.

parameters
is a list of transmission settings for a serial device driver. Do not include these settings for parallel device drivers, which are standardized. The transmission settings include baud rate, parity, stop bits, and bits per character.

FCBS

FCBS = x,y↵

FCBS specifies the number of file control blocks (FCBs) that MS-DOS can have open concurrently while file sharing is in effect. This command has no function while file sharing is not in effect.

x

is a number from 1 to 255 that specifies the total number of files that FCBs can have open at the same time. The default value is 4. Increase the value of x if you run a program that encounters errors because FCB files are closed by MS-DOS.

y

is a number from 0 to 255 that specifies the number of files opened by FCBs that are protected by MS-DOS if an application instructs FCBs to open more than **m** files simultaneously. MS-DOS protects the first **n** files opened by FCBs from being automatically closed. y must be less than or equal to x. The default value is 0.

If a program attempts to load more than x files, MS-DOS opens the new file and closes the file you have used least recently. The first y files, however, are protected from being closed.

If x is equal to y and a program tries to load more than x files, MS-DOS cannot close or open any files.

FILES

where **xxx** is a number between 8 and 255. There is a limit of 20 files per process (program). The default value is 8. This command specifies the number of files that can be opened concurrently using XENIX™-type file handles. (XENIX file handles are used by system calls 2F hex through 60 hex.) For each additional file above 8, the size of the resident portion of MS-DOS increases by 48 bytes per file. The amount of memory available for the application program is reduced by the number of bytes allocated to the number of opened files. **Note:** Specifying a value less than 8 might prevent some system operations, such as filtering and piping (see Chapter 5).

Your programs can read, write, or close any file by telling MS-DOS which file handle to use. When an application uses a file handle to open a file, MS-DOS makes a file control block in memory in an area set aside when MS-DOS is loaded. The size of the area, which controls the number of file handles that can be open concurrently, is determined by the FILES command.

FILES has no effect on the number of files that can be opened using FCBs.

LASTDRIVE

LASTDRIVE = x⏎

LASTDRIVE specifies the maximum number of drives that you can access. LASTDRIVE is useful if not all your drives are accessible during the booting process. For example, if your system maps in network drives after it is booted up, you can use LASTDRIVE to indicate the number of drives that will be accessible.

x

   is a letter that represents the last drive you can access. If you do not specify the last drive, the default is LASTDRIVE = E.

You must set LASTDRIVE equal to at least the number of drives on your system. Otherwise, the LASTDRIVE command is ignored in the configuration file.

SHELL

SHELL = [path]filename.ext [switches]⏎

The SHELL command is used in a CONFIG.SYS file to tell MS-DOS which command processor to load. The command processor is the first program to be run when you load MS-DOS. It translates your input into a series of steps for MS-DOS to perform. Because the command processor provides the complete "world" you see when you operate the computer, it is said to form a "shell" around the low-level operations of MS-DOS.

The default command processor is COMMAND.COM. If there is no CONFIG.SYS file in the boot directory, MS-DOS looks for a file named COMMAND.COM. If there is a CONFIG.SYS file containing a SHELL command, MS-DOS tries to load the command processor named in the SHELL command as the first program to be run.

path
> is the path through the directory hierarchy to the file containing the command processor. Include the drive name and subdirectory names where necessary.

filename.ext
> is the name of the file containing the command processor.

switches
> are any switches the command processor needs once it starts. If you are using COMMAND.COM, the /P switch is required for normal operation. The /D switch can be used as well; see COMMAND in Section 6.4.

The following command is the most common form of SHELL; it tells MS-DOS to load COMMAND.COM:

```
shell = command.com /p
```

Instead you could include this command in your CONFIG.SYS file:

```
shell = debug.com
```

With this SHELL command, the DEBUG utility will be loaded immediately after MS-DOS starts and will become the operating environment for your computer.

If you are a systems programmer you can replace COMMAND.COM with a command shell of your own. This shell could be graphical, menu-driven, or contain "shell variables" or a small programming language. Such alternatives are beyond the scope of this manual; designs and suggestions can be found in technical references and literature for the UNIX™ operating system.

If you replace COMMAND.COM with your own command processor, you must provide for interrupts 22, 23, and 24 hex, and must handle proper memory arena ownership and sizing, particularly during initialization.

6.4   Command Descriptions

This section presents descriptions of the syntax and examples of usage for the MS-DOS commands.

Assign Drive (ASSIGN)

ASSIGN [x = y...]⏎

ASSIGN is an external command that instructs MS-DOS to use a drive other than the one requested. In effect, physical drive X is converted to logical drive Y. When you enter this command, do not use colons after the drive name.

x

   is the physical drive.

y

   is the logical drive assignment.

When you use the ASSIGN command without any parameters, all drive assignments are reset to normal.

With the ASSIGN command, you can use application programs designed to use only drives A and B. For example, an application program that offers only A and B as choices can also be used with drive C.

The ASSIGN command should be used only in such application programs. ASSIGN must never be used with the PRINT command or in normal MS-DOS operations because this use can hide device type commands and programs that require actual disk operation. The DISKCOPY, FORMAT, and DISKCOMP commands ignore all drive reassignments.

When you enter the following command, procedures previously assigned to drive A are performed on drive B. For example, if you enter the command and then a directory command, MS-DOS displays a directory listing of the files located in drive B.

assign a = b⏎

The following command line causes procedures assigned either to drive A or to drive B to be performed on drive C:

assign a = c b = c⏎

If you specify ASSIGN without any parameters, all previous assignments are undone; physical drive A is logical drive A, physical drive B is logical drive B, and so on:

assign⏎

CAUTION: Do not use BACKUP, RESTORE, LABEL, SUBST, JOIN, or PRINT with an ASSIGNed drive.

Set Read-Only Attribute (ATTRIB)

ATTRIB [+ R] [-R] d:[path][filename[.ext]]⏎

ATTRIB is an external command that sets or resets the read-only attribute of a file. or displays the current setting of the read-only attribute.

+ R

sets the read-only attribute for the specified file.

-R

disables the read-only attribute for the specified file.

To allow file sharing on a network, ATTRIB + R forces the read-only mode even if the application software opens a file in read and write mode.

The following example enables the read-only attribute of the file ACCOUNTS.TXT in the current directory:

attrib + r accounts.txt⏎

Type the ATTRIB command followed by a filename to display the read-only attribute setting for that file. The following command displays the setting for the file ASSETS.TXT:

attrib assets.txt⏎

You can use wildcard characters to display the read-only attribute setting of more than one file. The following command displays the attribute setting for all files in the current directory:

attrib *.*⏎

Fixed Disk Backup (BACKUP)

BACKUP d:[path][filename[.ext]] d: [/A] [/D:mm-dd-yy]
[/L:[filename[.ext]]] [/M] [/P] [/S] [/T:hh:mm:ss]⏎

BACKUP copies files from the fixed disk to diskettes or another fixed disk, placing the files on the backup disk in a special BACKUP format. You can copy an entire disk or a specified group of files. Use only diskettes that have been formatted on your computer. To access the backup files for later use, you must use the RESTORE command. The source disk must not be write-protected because BACKUP marks the files with the archive attribute.

/A

adds files to the end of the diskette that is already in the drive. Use /A any time you want to save the current contents of the backup diskette; otherwise all pre-existing backup files are erased during the backup procedure. This option enables you to add new files to an existing backup diskette.

/D

backs up only those files with a date and time of creation later than the specified date. The DATE format varies according to the country; see Section 1.3 for details.

/L

makes a backup log entry in the file specified. If you omit the filename, BACKUP puts the log in the root directory under the filename BACKUP.LOG. The log contains the date and time of the backup as well as each filename and the number of the diskette that contains the file. If a log already exists, the new entries are added to the file.

/M

backs up only those files that are new or have been modified since the last backup. This option causes the system to ignore files that are unchanged since the last backup, so less time is required for the backup procedure.

/P

packs as many files as possible onto the diskette. If you use this switch, you may lose IBM BACKUP/RESTORE compatibility.

/S

backs up all files in all levels of subdirectories within the specified directory. With this option you can back up a specific directory tree without having to specify all the files and without backing up the entire fixed disk.

/T

backs up only those files modified after the time specified.

If the target drive is a fixed disk, BACKUP stores files in the \BACKUP directory. If the target drive is a floppy disk drive, BACKUP stores files in the root directory.

If you do not specify a pathname, the current directory is backed up. The directories and filenames are displayed as they are backed up. You are prompted to insert additional diskettes if they are needed.

Unless you use the /A option, BACKUP copies over all existing files on the backup diskette. Before doing so, the system prompts you to verify that you want to erase the diskette. To stop the BACKUP procedure and save the existing backup files, press Ctrl-Break in response to this prompt.

You do not need to use the /A option if you are backing up onto an empty diskette or want to completely overwrite an existing backup diskette with current versions of the files. **Do** use /A if you want to add files to the diskette without erasing the existing backup files.

You can use wildcard characters in the filename to back up particular files or groups of files. For example, the following command backs up all files in the current directory on drive C with the filename extension .DOC to the diskette in drive A. Since the /A option is not specified, any files already on the diskette are erased.

    backup c:*.doc a:⏎

The following command backs up all files in the current directory on drive C that have been modified or created (/M) since the last backup. They are backed up onto the diskette in drive A. Since /A is not specified, any files already on the diskette are erased.

   backup c: a: /m⏎

This command erases any files on drive B (no /A) and backs up all files in all drive D directories (/S) to the diskette in drive B:

   backup d:\ b: /s⏎

The following command backs up all drive C files (/S) modified or created (/M) after February 4, 1986 (/D) to drive B:

   backup c:\ b: /s /m /d:2-4-86⏎

This command adds (/A) all files (/S) on drive C that have been modified or created since the last backup (/M) to the backup diskette in drive A:

   backup c:\ a: /s /m /a⏎

The preceding command is recommended if you want to keep a continuous archive of backup files; previous backup files are not erased.

BACKUP sends an error level number to the batch command IF–ERRORLEVEL. (See the IF command for details on how you can use these error levels to control subsequent error-level processing.) The error levels set by BACKUP are:

0   Normal completion

1   No files were found to back up

3   Terminated by user (Ctrl-Break or Esc)

4   Terminated due to error

If you are sharing files, you can back up only those files you have access to.

CAUTION: Do not use BACKUP when a JOIN is in effect. Do not use BACKUP on a drive that is part of an ASSIGN or SUBST.

Ctrl-C Interrupt (BREAK)

BREAK [ON]↵

or

BREAK [OFF]↵

BREAK is an internal command that lets you use Ctrl-C as a system interrupt (BREAK ON) or without the interrupt effect (BREAK OFF) when MS-DOS is processing a program. When BREAK is ON, pressing Ctrl-C interrupts the current operation and redisplays the system prompt. Enter BREAK without parameters to check the status of the interrupt function. MS-DOS displays the following, where xx is "on" or "off":

```
BREAK is xx
```

ON tells MS-DOS to allow a Ctrl-C interrupt at any time during the processing of a program. OFF tells MS-DOS to allow a Ctrl-C interrupt only during the operation of the keyboard, printer, or an auxiliary device. The default is OFF.

Change Directory (CHDIR)

CHDIR [..][path]↵

CHDIR is an internal command that changes your position in the directory hierarchy or displays the name of the current subdirectory. You can abbreviate the command name to CD. CHDIR is also described in Chapter 3.

You use CHDIR to:

▶ Move to a subdirectory from any other directory.

▶ Move to a directory level above the current directory.

▶ Return from a subdirectory to the root directory.

▶ Display the name of the directory you are using.

CHDIR without parameters displays the directory you are currently using. For example, if you are in a directory called CMD in the default drive and specify

chdir↵

MS-DOS displays the following:

```
A:\cmd
```

Moves you to the directory above the one you are currently using. You can move up two directory levels by specifying

cd ..\\.. ↵

path

is the path through the directory system to the subdirectory level you want. Each directory name is the name of a directory created with the Make Directory (MKDIR) command.

For example, the following command looks for the EDFILES subdirectory three levels below the root directory:

chdir \\mgr1.nam\\cmd\\edfiles ↵

If you specify a directory name without the \\ parameter, MS-DOS looks for the directory name in the current directory. In the previous example, if you are in the CMD subdirectory and you want to access EDFILES, you can enter

chdir edfiles ↵

instead of the path \\MGR1.NAM\\CMD\\EDFILES. MS-DOS looks in the current directory, CMD, for the EDFILES subdirectory.

Entering \\ without a directory name refers to the directory at the top of the directory hierarchy, the root directory. This CHDIR command returns you to the root directory:

**chdir ** ↵

CAUTION: You should not use CHDIR on a directory that is part of an ASSIGN, a SUBST, or a JOIN.

Check Disk (CHKDSK)

CHKDSK [d:][filename.ext] [/F] [/V]↵

CHKDSK is an external command that scans the directory hierarchy of the diskette in the specified drive for consistency. If inconsistencies are found, error messages and a status report are displayed.

If you enter CHKDSK without parameters and if the diskette in the default drive contains errors, MS-DOS displays

```
Errors found, F parameter not specified.
Corrections will not be written to disk.

n lost clusters found in n chains
Convert lost chains to files (Y/N)?
```

If you respond with Y, MS-DOS tries to convert the lost chains to files, but cannot because you did not use the /F parameter in the command line. If you respond with N, MS-DOS displays the amount of memory that would be freed.

d:
is the name of the drive containing the diskette that you want to check.

filename.ext
is the name of the file you want to check. Use wildcard characters to refer to more than one file. For example, *.* reports extents (contiguous space on disk reserved for a file) for all files in the current directory after a normal consistency check is made. You can copy and rename files with many extents to restore them to a contiguous state.

/F

automatically reports to you when:

- ▶ There is an allocation error. CHKDSK fixes the error by adjusting the size.

- ▶ There is a disk error during the reading of the File Allocation Table (FAT).

- ▶ There is a disk error during the writing of the File Allocation Table (FAT).

- ▶ The file for the specified filename contains noncontiguous blocks.

- ▶ The first cluster (file attributes, password, allocation, extent, and statistical information) number is invalid and the file entry is truncated. CHKDSK fixes the error by releasing the allocation units assigned to that file.

- ▶ The command to change directories for the specified filename cannot be done because a subdirectory cannot be processed.

You must correct the error when CHKDSK encounters:

- ▶ An incorrect MS-DOS version. You cannot run this version of CHKDSK on MS-DOS versions lower than 3.1.

- ▶ Insufficient memory. Processing cannot continue; there is not enough system memory for CHKDSK to process the diskette.

- ▶ An invalid current directory. Processing cannot continue. Restart the system and reissue the CHKDSK command.

- ▶ A command to change directories cannot be made to the root directory. Processing cannot continue. The diskette is bad. Restart MS-DOS and issue the RECOVER command.

- ▶ Cross-linked files. The same data block is allocated to more than one file. Use COPY to copy the files you want to keep. Check the copies for accuracy, then delete the cross-linked files.

- ▶ A non-MS-DOS disk. You are prompted to respond yes (Y) or no (N) if you want CHKDSK to continue processing.

▶ Insufficient room in the root directory. You are prompted to erase the files in the root directory and reissue the CHKDSK command.

▶ An unrecoverable error in the directory. You are prompted to respond yes (Y) or no (N) to convert the directory to a file. If you respond Y, you can fix the directory yourself or delete it.

▶ Lost allocation units. Lost allocation units are called "orphans." Usually orphan allocation units occur when you do not close a file correctly (created, written to, but not closed). You can either free the orphan allocation units or recover them. A CHKDSK recover operation writes orphan allocation units to files in the root directory. If there is not enough room on the diskette to write all the orphan files, delete some files and run CHKDSK again.

/V

displays a trace (a record of the series of events as they occur) of the files and directories during the CHKDSK processing.

This example checks the diskette in drive B and issues a status report:

```
A>chkdsk b: ↵

  620544 bytes total disk space
   63488 bytes in 2 hidden files
    2048 bytes in 1 directories
  237568 bytes in 16 user files
  317440 bytes available on disk

  472976 bytes total memory
  458576 bytes free

A>_
```

CAUTION: Do not use CHKDSK with network drives or drives that are part of a SUBST or a JOIN.

Clear Screen (CLS)

CLS↵

CLS is an internal command that clears the screen and redisplays the system prompt.

Reload Command Processor (COMMAND)

COMMAND [path] [device] [/P] [/D] [/F] [/E:n] [/C string]↵

COMMAND.COM is an external program that translates the commands you enter into a series of instructions telling MS-DOS what to do. COMMAND.COM is usually loaded by the operating system at bootup time (see the SHELL command in Section 6.3). COMMAND displays the system prompt, prompts for date and time, asks MS-DOS to load external programs, and contains the instructions to perform all the internal MS-DOS commands.

You can enter COMMAND interactively at the system prompt to load another copy of the command processor. The copy of COMMAND originally loaded by MS-DOS will load a second copy of COMMAND, which can then load further copies of COMMAND. Thus you can run copies of COMMAND inside copies of COMMAND inside copies of COMMAND. The depth to which copies of COMMAND can be "nested" in this way depends only on the size of your computer's memory.

The ability to run several copies of COMMAND is most often used in batch files to nest other batch files. Ordinarily if a batch file contains the name of a second batch file, when the second batch file finishes running, the batch job is over. That is, control does not return to the first batch file so that it can continue. If a second copy of COMMAND is started, however, the name of the second batch file can be passed to it (see the /C switch description). Then when the second batch file

finishes running, the second copy of COMMAND finishes also, and the first batch file regains control and continues execution.

Whether COMMAND is started by MS-DOS at system bootup or by you at the system prompt, there are several options which can be used to change the way COMMAND operates.

path
is the path through the directory system to the COMMAND.COM file. The path includes the drive name and any subdirectory names where necessary. No path is necessary if the file is in the root directory on the default drive.

Although part of COMMAND is always in memory once it is loaded, a major piece of it is "thrown away" when one of your application programs requires a large amount of memory. The path you specify allows COMMAND to find itself on disk and reload the part which was overwritten by your application. If you do not specify a path, COMMAND uses the one specified in CONFIG.SYS or in the root directory from which the copy of COMMAND that MS-DOS found at bootup was loaded.

device
is the name of the TTY device COMMAND uses for input and output. If you do not specify a device, COMMAND uses the CON device (the default keyboard and screen).

/P
makes this copy of COMMAND the "permanent" one. Usually /P is used only with the SHELL command in CONFIG.SYS. If you have loaded several copies of COMMAND, you can exit to the previous (outer) copy of COMMAND using the EXIT command. If /P was specified when you started the current copy of COMMAND, however, this copy is the permanent one and there is no other command processor "outside" it. This is always the case with the first command processor that MS-DOS loads. The /P switch also makes COMMAND run any AUTOEXEC.BAT and CONFIG.BAT files and ask for the date and time (unless you turn this prompt off with the /D switch).

/D

tells COMMAND not to prompt for the date and time. This switch is useful if you have a real-time clock installed which sets the date and time automatically.

/F

causes any disk errors which result in an Abort, Retry, Ignore message to be aborted. This switch is useful in batch files to allow batch processing to continue when an error is encountered instead of making the system wait for a response.

/E:n

sets the size of the environment in bytes. The environment is a series of text messages passed from program to program which allows applications to gain information about the current configuration of the system. (See the SET command in this chapter for more information.) Different applications store different types of information in this area. If you do not specify the size, a default environment of approximately 384 bytes is allocated. This amount is generally adequate for most applications.

/C string

tells the command processor to execute the command in "string" and then exit, that is, return to the previous (outer) copy of COMMAND. The /P and /C switches are mutually exclusive because there is no outside processor to exit to after the /C string has been executed. In order to nest batch files, as described above, place the following command in your first batch file:

command /c batchtwo

BATCHTWO is the name of a second batch file to be executed. After BATCHTWO finishes, the second copy of COMMAND is exited and control returns to the first batch file.

Compare Files (COMP)

COMP [d:][path][filename[.ext]] [d:][path][filename[.ext]] ↵

COMP is an external command that compares the contents of one file
to the contents of another file.

d:path\filename.ext

The first parameter is the primary file, and the second is the sec-
ondary file. These files can be on the same drive or on different
drives, and in the same directory or in different directories.

Wildcard characters can be used in either of the filenames and exten-
sions. If you specify only the drivename and the path, and do not
specify a filename, MS-DOS assumes a filename of *.*.

To compare identical files in two different directories on two different
drives, enter a command similar to this:

comp a:\dir1*.c b:\dir2\dir3 ↵

Each file located in the directory DIR1 on drive A and having the .C
extension is compared to a file with a matching filename and extension
and located on drive B in the path DIR2\DIR3. As each file is com-
pared, its path and filename are displayed.

If the files being compared are of different lengths, MS-DOS displays the
message "Files are of different sizes" and no comparison takes place.

If information in the two files being compared doesn't match, the dif-
ferences are displayed as follows:

```
Compare error at offset nnnnnn
File 1 - xx
File 2 - xx
```

nnnnnn is the number of bytes from the start of the file to the byte at which the files differ.

Up to ten compare errors can be reported for any given file. If there are more than ten errors, MS-DOS assumes that further comparison is useless, and it goes on to the next file. If no compare errors are found, the message "Files compare OK" is displayed.

Certain application programs create directories that always record file size in multiples of 128. Therefore, the number of information bytes in a file might be less than is indicated in the directory. In such cases, the information bytes in the two files might compare, but the COMP command might find differences in the non-usable (filler) portion of the file that follows the information bytes. If this occurs, MS-DOS displays the message "EOF mark not found."

Concatenate and Output Files (CONCAT)

CONCAT [files or directories]⏎

CONCAT is an external command that concatenates (combines) a list of files and displays their contents. The concatenated files are sent to the standard output device, the console, but are not saved on disk unless you include the > redirection symbol to send the output to a disk file, as shown in the first example that follows.

files or directories
   is a list of subdirectory names and/or filenames to be operated on. If you do not specify files or directories, input is taken from the keyboard or from redirected input.

For example, the following command concatenates the TOD.ASM and ATC.ASM files in the current directory. The > symbol sends the combined files to the file BACKUP.

concat tod.asm atc.asm > backup⏎

You can also use CONCAT to create files. This command takes any following keyboard input (up to Ctrl-Z and Enter) and outputs it to the disk file JUNK:

    concat > junk↵

Copy Files (COPY)

**COPY [path]filename.ext [/A] [/B] [+ [path]filename.ext [/A] [/B]...]
[path][filename.ext] [/A] [/B] [/V]↵**

COPY is an internal command that copies one or more files from a source file to a destination file. The source file is the file you want to copy. The destination file is the file to which you want to copy your source file. If you do not specify a destination drive, your file is copied to the current or default drive.

path
is the path through the directory hierarchy to the source or destination file. The path includes the drive name and subdirectory names as necessary.

filename.ext
is the name defined when the file was created. It can also be the name of a new file created during the COPY operation.

To copy a file to the default drive, specify the drive containing the source file and the filename. After this COPY operation, the copy in the default drive has the same name as the source file.

For example, this command copies the file ASSETS from drive B to the default drive (drive A) and gives it the same name:

    copy b:assets↵

The copy is made to the current directory on the default drive.

To copy the file from the default drive to another drive, specify the name of the file to be copied and the name of the drive. For example, this command copies the file ACCOUNTS in the default drive to drive B and gives it the same name:

copy accounts b:⏎

Using only filenames as parameters, you can:

▶ Copy a source file from another drive to a destination file on the default drive and keep the same filename:

copy b:chkdisk.bat⏎

▶ Copy a source file from the default drive to a destination file with the same name on a different drive:

copy chkdisk.bat b:⏎

▶ Copy a source file to a destination file on the same drive and rename the file:

copy chkdisk.bat newfile⏎

The CHKDISK.BAT file is copied on the default drive and is renamed NEWFILE.

▶ Copy a source file from the default drive to another drive and rename the file:

copy chkdisk.bat b:newfile⏎

▶ Copy a source file from a specified drive to the default drive and rename it:

copy b:chkdisk.bat newfile⏎

The file CHKDISK.BAT on drive B is copied to the default drive and is renamed NEWFILE.

+

concatenates files (links them together). MS-DOS adds a file or files to the end of the first specified file and puts the result in either the first file or the destination file that you name. For example, this command adds the file PLUS.FIL to the end of ONE.FIL and puts the result in ONE.FIL:

copy one.fil + plus.fil ↵

The next two examples put the result of the concatenated files into another file. In this example, the contents of the file PLUS.FIL are added to the end of ONE.FIL and the result is renamed BIG.FIL:

copy one.fil + plus.fil big.fil ↵

In the next example, the contents of the file NEXT.FIL on drive B are added to the end of the file PLUS.FIL on the default drive, and the resulting file is added to the end of ONE.FIL on the default drive. The destination file is called BIG.FIL.

copy one.fil + plus.fil + b:next.fil big.fil ↵

You can concatenate files using wildcard characters. This command searches for and combines all the files with extensions .LST and .REF. The result of the concatenation is put into COMBIN.FIL. FILE1.LST is combined with FILE1.REF and the result is put in COMBIN.FIL, and so on.

copy *.lst + *.ref combin.fil ↵

The following command combines FILE1.LST with FILE1.REF to create FILE1.FIL. ABC.LST combines with ABC.REF to create ABC.FIL, and so on.

copy *.lst + *.ref *.fil ↵

This command can produce an error if ALL.LST already exists:

copy *.lst + all.lst ↵

As each source filename is found, MS-DOS compares the name with the destination filename. If the names are the same, as in the preceding command, MS-DOS skips the source file and displays the message "Content of destination lost before copy." Further concatenation continues normally.

You can, however, add all the files with extension .LST to the file ALL.LST:

    copy all.lst + *.lst↵

/A

specified with the source file causes the file to be processed as an ASCII (text) file. The file is copied to the first end-of-file character (Ctrl-Z). The Ctrl-Z and the rest of the file are not copied. Suppose you enter the following command:

    copy file.txt /a memo.txt↵

If there is a Ctrl-Z embedded in FILE.TXT, the Ctrl-Z is removed when FILE.TXT is copied to MEMO.TXT. MEMO.TXT contains only one Ctrl-Z, the last character of the file.

/A specified with the destination file puts a Ctrl-Z at the end of the file. /A is the default when files are concatenated using COPY.

/B

specified with the source file copies the entire file to the physical end-of-file defined by the directory file size. For example, enter

    copy b:new.fil /b cop.fil /a↵

The /B on the source file keeps the end-of-file mark (Ctrl-Z), and the /A on the destination file inserts a Ctrl-Z. A /B specified with the destination file stops the Ctrl-Z from being placed at the end of the file.

By default, MS-DOS concatenates files as ASCII (text) files. A Ctrl-Z (1A in hexadecimal) in the file is interpreted as an end-of-file mark.

When you want to combine binary files, use /B. /B forces the COPY command to use the physical end-of-file defined in the directory. /B remains in effect until MS-DOS encounters another /A or /B. For example, entering

```
copy /b bin.fil + cop.fil↵
```

concatenates COP.FIL to the end of BIN.FIL and places the result into BIN.FIL as a binary file. Note that the COPY command is not a linker; object modules cannot be combined into an .EXE or a .COM file.

/V

tells MS-DOS to verify that the sectors written on the destination disk are recorded correctly. Use /V to check that critical data is copied accurately. This switch has the same effect as the VERIFY = ON command.

When you use /V, COPY performs more slowly because extra time is needed to verify the COPY operation.

You can also use reserved device names (see Section 2.2.4) in a COPY command. For example, you can copy your keyboard input into a disk file with COPY CON:

```
A>copy con text.fil↵
This is a text file↵
Each time you press the Enter key, a line is
created in the file↵
The last line ends with a Ctrl-Z (and Enter)↵
^Z↵

A>_
```

After you press Ctrl-Z (^Z) and Enter, the copy ends and the file is saved as TEXT.FIL.

Change Input/Output Devices (CTTY)

CTTY [devicename]⏎

CTTY is an internal command that controls the console device: the keyboard and screen. If you enter CTTY without specifying a device name, the current console device is displayed.

devicename
  is the name of a device driver. You must use a valid device name for a device driver that was loaded in the CONFIG.SYS file (using the DEVICE command). CONFIG.SYS, on your system diskette, installs the device drivers. CON is the console I/O device for CTTY.

To prevent confusion between device names and filenames, precede device names with \DEV\. For example, CTTY = CON is equivalent to CTTY = \DEV\CON.

To select an alternate device driver, type its device name with CTTY; for example:

ctty remote⏎

6

Current Date (DATE)

DATE [date]⏎

DATE is an internal command. MS-DOS records the date in the current directory for every file that you create or change. If you specify DATE without parameters, MS-DOS displays the day and the numeric representation for the month, the day, and the year last set. You are also prompted for the new date, as follows:

```
Current date is Sun 6/22/86
Enter new date:_
```

If the date is incorrect, enter the new date. If you do not want to change the displayed date, press Enter. You can also enter the DATE command with a date to change the date known to the system. (If your system has a real-time clock, you might not be able to change the date using the DATE command.)

For example, to enter the date as June 25, 1986, type

date 6/25/86⏎

The valid date format for your operating system is determined by the COUNTRY code in the CONFIG.SYS file. See Section 1.3 for details.

The date format consists of the following:

mm

   is the number for the month (1 through 12).

dd

   is the number for the day (1 through 31).

yy

   is the number for the year (80 through 99, or 1980 through 2099).

Separate the month, day, and year with hyphens (-) or slashes (/). If you use any symbols other than these, or if you use letters instead of numbers, the following message and prompt appear:

```
Invalid date
Enter new date:_
```

You can change the date from the keyboard or from a batch file. However, if you are using an AUTOEXEC.BAT file during MS-DOS startup that does not contain the DATE command, MS-DOS does not prompt for the date. For this reason, you might want to include the DATE command in that file.

Delete Files (DEL)

DEL [path]filename.ext⌐

DEL is an internal command that deletes all files matching the speci-fied filename.

path
    is the path through the subdirectory system to the file you want to delete. Include the drive name in the path if the file is not on the default drive.

Enter DEL *.* to delete all the files. MS-DOS displays

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

You must type Y or N followed by Enter to confirm or retract your DEL *.* command.

This command deletes the file EXAMP.TXT from the TEXT sub-directory on drive B:

del b:\bin\text\examp.txt⌐

The following command deletes all the files with .LST file extension from the current directory on the default drive:

del *.lst⌐

CAUTION: Use DEL carefully with files on a drive that is part of an ASSIGN, a SUBST, or a JOIN because you might affect drives other than the one you specify.

Display Directory (DIR)

DIR [path][[filename][.ext]] [/P] [/W] ↵

DIR is an internal command. DIR without parameters lists all the files in the current directory on the default drive.

path
    is the path through the subdirectory system to the subdirectory you want to list, or to the file(s) whose existence you want to verify.

filename
    lists all the files with the given name, regardless of their extension.

filename.ext
    lists only those files from the root directory with the given name and extension.

/P

    stops listing the directory at one page or screenful of information. MS-DOS prompts you to strike any key to continue viewing the directory.

/W

    displays five filenames per line across the width of the screen. Only the filenames are displayed.

DIR lists the filenames and their sizes in bytes, plus the time and the date they were last modified, for all the parameters except /W.

You can use the ? and * wildcard characters (see Section 2.3) in the place of parts of a filename. For example:

dir	is the same as	**dir *.***
dir filename	is the same as	**dir filename.***
dir .ext	is the same as	**dir *.ext**

The following example shows a sample directory listing. The backslash indicates the root directory on drive A. When your current or default directory is the root directory on drive A, you can type the directory command without specifying a drive name.

```
A>dir↵

Volume in drive A is JANUARY
Directory of A:\

STATUS      DAT      10368    1-23-86    2:12p
COST        DAT         31    1-31-86   11:06a
STATUS      BAK      10334    1-16-86    3:32p
CALENDAR    JAN       4068    1-12-86   12:30p
MEMOS            <DIR>           1-03-86   12:43p
LETTERS          <DIR>           1-10-86   10:04a
        6 File(s)    141516 bytes free

A>_
```

The next example shows the same directory listing using the /W option.

```
A>dir/w↵

Volume in drive A is JANUARY
Directory A:\

STATUS   DAT COST    DAT STATUS BAK CALENDAR JAN   MEMOS
LETTERS
        6 File(s)    141516 bytes free

A>_
```

Diskette Comparison (DISKCOMP)

DISKCOMP [side to side] [/1] [/8]↵

DISKCOMP is an external command that compares the contents of a diskette in one drive with the contents of a diskette in another drive. DISKCOMP always compares two entire diskettes to each other.

side to side
> is an optional phrase naming the drives containing the names of the diskettes that you want to compare.

/1
> compares only the first side of two diskettes even if you are using double-sided diskettes.

/8
> compares only 8 sectors per track even if you are using 9-sector-per-track diskettes.

You can use DISKCOMP to verify that the DISKCOPY command has accurately copied the contents of one diskette onto another diskette.

You should not use DISKCOMP when the COPY command was used to copy a diskette. Although the contents of the two diskettes created with COPY might be identical, the arrangement of the data on each diskette might not be identical. In this case, DISKCOMP indicates that the corresponding tracks on the diskettes do not compare even though all files might have been accurately copied.

When you enter DISKCOMP, one or two drives can be specified. If you specify only one drive, DISKCOMP performs a single-drive comparison. Use this form with a system having only one diskette drive and a fixed disk drive. MS-DOS prompts you to insert diskettes at the appropriate time. After giving the prompt, MS-DOS waits for you to press any key.

If DISKCOMP finds a discrepancy between the diskettes, you see this
message:

```
Compare Error(s) on
Track nn, Side x
```

where **nn** is the track number and **x** is the side of the diskette (0 or 1)
on which the discrepancy was found.

If the diskettes compare exactly, the message "Diskettes compare OK"
is displayed.

If you do not specify any parameters, a single-drive comparison is
done using the default drive. If the second drive name is omitted, the
default drive becomes the second drive. If the default drive is specified
as the first drive name, a single-drive procedure is used.

When you attempt to compare diskettes on a single-drive system, all
prompts are for drive A, regardless of the parameters specified.

Diskette Copy (DISKCOPY)

DISKCOPY [source drive:] [target drive:] [?] [/S] [/K] [/R] [/F] [/V] [/O overflow drive:]⏎

DISKCOPY is an external command that copies the entire contents of the diskette in the source drive to the diskette in the target drive. DISKCOPY automatically determines the format of the diskette, the number of sectors per track (8, 9, 15, or variable), and the number of sides to copy, based on the layout of the source diskette. If the target diskette is unformatted, DISKCOPY formats it before writing data to it.

source drive
> specifies the drive containing the diskette you wish to copy. Enter the drive letter followed by a colon.

target drive
> specifies the drive containing the diskette that will be the copy. Enter the drive letter followed by a colon. **CAUTION:** If the target diskette already has data on it, the data is erased during the copying process.

If you omit one of the two drive specifiers, DISKCOPY uses the drive you do specify as the source drive. The target drive is the default drive. If the drive specifier you enter is the same as the default drive, a single-drive copy is performed; DISKCOPY prompts you to insert the target and source diskettes alternately until the copy is completed.

If you do not specify any drive in the command, a single-drive copy is performed as described in the preceding paragraph. The default drive is both the source and target drive.

?
> displays two screens of help information describing DISKCOPY syntax and switches. No copy is performed.

/S

causes DISKCOPY to "sign on" with its name, version number, and date of creation.

/K

causes DISKCOPY to begin copying immediately without prompting you ("kills" prompts); when the copy is finished DISKCOPY exits to the MS-DOS prompt. ERRORLEVEL is set to zero if no errors occurred; otherwise it is set to one of the error codes described at the end of this section. If the source and target are the same, DISKCOPY overrides the /K switch and prompts you to insert the source or target diskette as necessary. This option is useful when DISKCOPY is called from a batch file (you must insert diskettes in the source and target drives before you process the batch file).

/R

causes DISKCOPY to exit to the MS-DOS prompt after copying if no errors occurred. If errors occur, when the copy is complete DISKCOPY prompts you:

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

This option allows you to insert new diskettes and repeat the DISKCOPY. If you respond N (no) to this prompt, DISKCOPY exits to the MS-DOS prompt with ERRORLEVEL set to one of the codes listed at the end of this section. If you use both /R and /K, DISKCOPY does not prompt you at all unless an error occurs.

/F

forces formatting of the target diskette whether it is already formatted or not. This option is useful in multiple-computer environments when you want to be sure that a diskette was formatted on the same computer you will be using.

/V

causes DISKCOPY to display messages ("verbose") describing each step of the copy process. DISKCOPY displays the numbers of the sectors it is currently reading and writing. While formatting a target diskette, DISKCOPY displays the number of the track currently being formatted.

/O overflow drive

provides an easy way to copy diskettes on a fixed disk computer that has only one disk drive. DISKCOPY loads the contents of your source diskette into memory. If the amount of memory in your computer is not sufficient to hold the entire source contents, DISKCOPY opens an overflow file on the fixed disk volume speci-fied by the overflow drive letter. Once all the source information has been read, DISKCOPY prompts you to insert a target diskette, and writes out all the data it has stored to the diskette, formatting if necessary. Once this process is complete, and if you did not specify the /K or /R options, DISKCOPY prompts you:

```
Copy another diskette using stored source
information? (Y/N)
```

If you respond Y (yes), you are prompted to insert a new target diskette, and DISKCOPY writes out the stored source information to this diskette. DISKCOPY reads the source information only once, and you can make multiple copies of a diskette.

With the /O option the source and target drives must be the same and the overflow drive specifier must be a volume on a fixed disk. Otherwise an error occurs. Specify this option last in the command.

CAUTION: You must determine whether the fixed disk has enough space to store the contents of the source diskette. DISKCOPY does not verify the amount of space available.

Unless prompting is suppressed, DISKCOPY prompts you when the copy is finished:

```
Copy complete.
Copy another? (Y/N)
```

If you respond N (no), DISKCOPY returns you to the MS-DOS prompt. If you respond Y (yes), DISKCOPY prompts you to insert source and target diskettes as required based on the drives you originally specified.

DISKCOPY cannot change the format of a diskette while copying; single-sided diskettes cannot be changed into double-sided diskettes or vice versa. To make format changes, use FORMAT to create a target diskette with the format you require and then use COPY *.* to move all the files to the new diskette.

When you first begin working with a new diskette, the data in a particular file is in the same general area of the diskette. But after you create and delete files over a period of time, available space is scattered over the diskette, and space can no longer be allocated sequentially. Pieces of your files are located in different areas of the diskette, and disk performance becomes slower.

This slowdown is known as fragmentation. DISKCOPY makes an exact copy of your source diskette; that is, the copy is equally fragmented. If you suspect that your diskette is fragmented, use FORMAT to format a new diskette and then use COPY *.* to copy your files to it. The scattered pieces from the source diskette are written sequentially to the new target diskette, and diskette performance should be greatly improved.

DISKCOPY uses as much memory as is available to hold the information being copied; thus the length of time for a diskette copy varies on different computers. When you use the /V option, the number of sectors displayed as DISKCOPY reads and writes varies depending on the number of memory-resident programs, device drivers, and buffers installed.

If an error occurs during the diskette copy, an error code is returned to MS-DOS which can be retrieved in a batch file by examining the ERRORLEVEL variable. The error codes are defined as follows:

0   No errors occurred.
1   Bad command line, bad drive specifiers, or bad switches.
2   Source drive invalid. Is a fixed disk or does not exist.
3   Destination drive invalid. Is a fixed disk or does not exist.
4   Error reading source diskette.
5   Error writing target diskette.
6   Not enough memory for copying.
7   Error writing overflow file.
8   Error reading overflow file.
9   Error opening overflow file.
10 Format error.
11 Other error.

Echo Messages to Display (ECHO)

ECHO ON↵

   or

ECHO OFF↵

   or

ECHO message↵

ECHO is an internal command for batch files that controls the display of batch file commands and comments during batch file processing. ECHO without parameters specifies the current setting (ON or OFF).

ON

    displays all batch commands and remarks (REM) during batch file processing. ECHO ON is the default.

OFF

    turns off display of the commands in the batch file. The command (such as A > ECHO OFF) is displayed, but any batch commands following this command are not displayed.

message

    is a string of alphanumeric characters that you want displayed during batch file processing.

    You can enter ECHO followed by two or more spaces to produce a blank line. This command is useful for creating an evenly spaced message on the screen.

For example, enter the following commands:

copy con greet.bat↵
echo Welcome aboard!↵
echo Enter your name.↵
^Z↵

These commands copy the two ECHO command lines to a batch file called GREET.BAT. A Ctrl-Z (^Z) closes the file and returns you to the MS-DOS system prompt. When you type GREET, MS-DOS displays

```
A>echo Welcome aboard!
Welcome aboard!
A>echo Enter your name.
Enter your name.
```

You can include ECHO OFF in a batch file to suppress just the display of the batch commands. And, in the same file, you can specify ECHO with a message to display that message during batch processing. For example, you can add ECHO OFF to the GREET.BAT file, as follows:

copy con greet.bat⏎
echo off⏎
echo Welcome aboard!⏎
echo Enter your name.⏎
^Z⏎

Then when you enter GREET from the keyboard, MS-DOS displays only the messages following each ECHO command:

```
A>echo off
Welcome aboard!
Enter your name.
```

Erase Files (ERASE)

Refer to the DEL command (delete files).

Convert Executable Files to Binary Files (EXE2BIN)

EXE2BIN [path]filename.ext [[path][filename[.ext]]] ↵

EXE2BIN is an external command that converts linker-generated .EXE (executable) files to binary (.COM) format. This procedure saves disk space and provides faster program loading.

The resident part of the .EXE file (the actual code and data) must be less than 64K.

The input file is converted to .COM format (memory image of the program) and placed in the output file. The input file must be in valid .EXE format produced by the linker. There can be no STACK segment.

path
>   is the path through the directory hierarchy to the file that you want to specify as the input file or the output file.

filename.ext
>   is the name of the input file. It can also be the name of the new file converted with EXE2BIN. If you do not specify a file extension with the input filename, it defaults to .EXE. If you do not specify an output filename, the input filename is used. If you do not specify a file extension in the output filename, the new file's extension is .BIN.

Two kinds of conversions are possible, depending whether the initial CS:IP (Code Segment:Instruction Pointer) is specified in the .EXE file:

▶ If CS:IP is not specified in the .EXE file, a pure binary conversion is assumed. If segment fixups are necessary (for example, the program has instructions requiring segment relocation), you are prompted for the fixup value. This value is the absolute segment at which the program is to be loaded. You can use the resulting program only when loaded at the absolute memory address specified by a user application. The command processor cannot load the program correctly.

▶ If CS:IP is specified as 0000:100H, it is assumed that the file is to be run as a .COM file with the location pointer set at 100H by the assembler statement ORG; the first 100H bytes of the file are deleted. No segment fixups are allowed because .COM files must be segment relocatable.

Once the conversion is complete, you can rename the resulting file with a .COM extension. Then the command processor can load and run the program in the same way as the .COM programs supplied on your MS-DOS diskette.

Exit COMMAND.COM (EXIT)

EXIT⏎

EXIT is an internal command. You can use this command to return from COMMAND.COM to the program or command processor you were using previously.

When MS-DOS is processing a program that allows you to call the COMMAND.COM program, you enter

command⏎

to invoke COMMAND.COM. You can now use any of the internal commands. To return to the previous program, type

exit⏎

Fast Expression Search (FGREP)

FGREP [options] expression [files or directories]⏎

FGREP is an external command that searches through a list of files for files that contain a specific text string. FGREP outputs the lines that contain the text string and the names of the files. Options allow you to search subdirectories, list only filenames, and list the number of lines that match. **options** can be any of the following:

/^

matches only text at the beginning of a line.

/$

matches only text at the end of a line.

/B

makes output buffered (the default for disk output).

/C

prints only a count of the lines containing matching text.

/I

ignores uppercase/lowercase when matching the expression.

/L

also prints line numbers for matching text in files.

/N

prints only the filenames of files with matching text.

/R

recursively searches subdirectories of the directories in the [files or directories] list. If you do not include this option, FGREP searches only the files you name in the command, as well as all files in any directories named in that command.

/U

makes output unbuffered (the default for terminal output).

/V

reverses the match—prints all lines except those with a matching expression.

/W

matches only text that is a word (or two words) not preceded by a letter, digit, or underscore.

expression

is the text string to search for. The expression cannot contain spaces or wildcards. Specify this string exactly as you want it to appear in the files—that is, without delimiters such as quotes. For example, this command searches for "jack" in the file CHAPT1.DOC on drive A:

fgrep jack a:chapt1.doc ↵

If the expression contains spaces, hyphens, or MS-DOS command-line characters (such as < or >), you must enclose the string in double quotes. For example, this command searches all files with the .PAS extension for "a jack-in-the-box", an expression containing spaces and hyphens:

fgrep "a jack-in-the-box" *.pas ↵

files or directories

is a list of one or more subdirectory names and/or filenames which can include wildcard characters. This list specifies the domain on which the FGREP command will operate. If no files are given, FGREP accepts keyboard input or redirected input to search.

For example, this command ignores uppercase/lowercase (/I) and prints a list of files and their lines that begin (/^) and end (/$) with (i.e., consist only of) the string "proc". Files in the directory C:\SOURCES and files in the D:\PROJECT directory with the .ASM file extension are searched.

fgrep /i^$ proc c:\sources d:\project*.asm ↵

This command acts as the previous example, except that any subdirectories (/R) of C:\SOURCES and their subdirectories will also be searched:

fgrep /ri proc c:\sources d:\project*.asm ↵

This final example outputs the names of files (/N) which have lines that do not contain (/V) the word "priority". FGREP searches the \MEMOS subdirectory on drive C, and all files with the .NOT extension in the root directory on drive D.

fgrep /vn priority c:\memos d:*.not ↵

Find Alphanumeric Characters (FIND)

FIND [/C] [/N] [/V] "string" [path]filename.ext... ↵

FIND is an external command for filtering data. This command searches a file to find a specified string of alphanumeric characters and displays all the lines on the screen that contain the specified string.

/C
displays only the number of matches with the string. The lines that match the string are not displayed.

/N
displays the relative number of the line(s) that match the string. FIND displays the relative line number, then the line.

/V
displays the lines that do not contain the string.

"string"
is a string of alphanumeric characters that you want to find. The characters must be enclosed in quotes. If they are already enclosed in quotes, use double quotes to specify the characters as the string.

path

is the path through the directory hierarchy to the file(s) to be searched for the string.

filename.ext

is the name of the file or files to be searched for the specified string.

The following example searches a mail list file for "Washington, D.C." You can create MAIL.LST by entering

copy con mail.lst ↵
George Washington, Washington, D.C. ↵
Charles de Gaulle, Paris, France ↵
Pierre Trudeau, Montreal, Canada ↵
Abraham Lincoln, Washington, D.C. ↵
Margaret Thatcher, London, England ↵
Ronald Reagan, Washington, D.C. ↵
^Z ↵

Then enter

find "Washington, D.C." mail.lst ↵

On the first line MS-DOS displays a dashed line, the path to the file if one was specified, and the name of the file:

```
----------mail.lst
```

MS-DOS then displays every line that contains the string "Washington, D.C."

If MS-DOS cannot locate the FIND command, MS-DOS displays the message "Bad command or filename." FIND is an external command and might be in another subdirectory. Enter a PATH command giving the location of FIND.

If MS-DOS displays the message "FIND: File not found mail.lst", the MAIL.LST file might be in a subdirectory. You need to specify the path to the file.

The FIND command can also filter data from an input source and pipe it to an output source (see Section 5.4). For example, you can find the characters "Washington, D.C." and pipe them to an output file called USPRES.DAT:

find "Washington, D.C." mail.lst I sort > uspres.dat⏎

The I tells MS-DOS to use the output from the FIND command as input to the SORT command.

Use the DIR command after you have piped data to verify that the file is in the directory that you want. If it is not, use the COPY command to put the file in the correct directory.

For Each One Repeat MS-DOS Command (FOR–IN–DO)

FOR %%variable IN (set) DO command %%variable⏎

FOR is an internal command used in batch files. It can also be used interactively. FOR repeats the specified MS-DOS command for each member in the specified set.

%%variable

is a dummy variable, where "variable" can be any character except the numbers 0–9 (which are used for the replaceable batch parameters, %0–%9). %%variable is replaced by each sequential member in the specified set. The variable specified at the beginning of the command line must be the same as the last variable in the command line. For example, if you specify %%A for the first variable parameter and %%F for the last variable parameter, MS-DOS displays the message "Entry error."

IN

tells MS-DOS to sequentially replace %%variable with each sequential member in the specified set.

(set)

can be any alphanumeric characters, including the replaceable batch parameters %0–%9. The set contains more than one member and must be enclosed in parentheses:

(mgr1 mgr2 mgr3)

You cannot specify a path in the set.

DO

tells MS-DOS to process the command that follows.

command

is an MS-DOS command.

For example, with FOR you can create a batch file that makes a new directory, changes the directory, and makes subdirectories:

```
copy con mgrdir.bat↵
mkdir %1↵
chdir %1↵
for %%a in (%2 %3 %4) do mkdir %%a↵
^Z↵
```

To process the file, enter the batch file name (MGRDIR) and the subdirectory names you want to create:

```
mgrdir managers mgr1 mgr2 mgr3↵
```

MANAGERS replaces %1, MGR1 replaces %2, MGR2 replaces %3, and MGR3 replaces %4. %%A is replaced by the first member of the set, which is MGR1. The command then tells MS-DOS, "FOR MGR1 IN (MGR1 MGR2 MGR3) DO MKDIR MGR1." MGR1 (%%A) is then replaced by MGR2, and MGR2 is replaced by MGR3.

MS-DOS displays each batch command as it executes:

```
A>mkdir managers
A>chdir managers
A>for %%a in (mgr1 mgr2 mgr3) do mkdir %%a
A>mkdir mgr1
A>mkdir mgr2
A>mkdir mgr3
A>_
```

Format Disk (FORMAT)

FORMAT [d:] [/S] [/1] [/8] [/V] [/B] [/4] ↵

FORMAT is an external command that formats a disk for use with MS-DOS files. The FORMAT command sets up the directory tracks and sectors on a diskette or fixed disk.

CAUTION: If you enter FORMAT without specifying a drive, the command formats the default drive. If you are logged on to fixed disk drive C and you enter FORMAT without a drive specifier, FORMAT reformats the entire fixed disk, **destroying the data** stored there. Always specify a drive, even when you intend to format the default drive.

A disk must be formatted before files can be stored on it. Usually, a disk is formatted only once: when it is new. When you format a disk, any information stored on it is destroyed. A disk formatted under an operating system that is not compatible with MS-DOS must be re-formatted before you can store MS-DOS files on it.

If portions of a disk are defective, they are not formatted, and the message "xxxxxxx bytes in bad sectors" is displayed along with other format data. If any bad sectors are found, there is less available storage space on the disk, but unless the defect is on track 0, you can still use the disk.

d:
specifies the drive to be used for formatting the disk.

/S
causes the MS-DOS system files to be copied onto the disk once it has been formatted. A diskette containing the operating system files must be in the default drive.

/1

creates a single-sided diskette on a double-sided drive. If you do not specify the /1 switch, FORMAT formats the diskette on both sides if you use a double-sided drive.

/8

formats the disk with eight sectors per track instead of the default of nine sectors per track. If the drive is double-sided, both sides of the disk are formatted with eight sectors per track. If the drive is single-sided, or if you specify the /1 switch, only one side of the disk is formatted. When you use the /8 option, FORMAT creates 9 sectors per track, but tells MS-DOS to use only 8 sectors per track.

/V

lets you assign a name to a disk when you format it. The name is displayed whenever you use the DIR command. If you specify /V, you are prompted to enter a volume label from one to eleven characters. If you do not want a volume label, press the Enter key.

/B

creates an eight-sector-per-track disk with space reserved for the operating system files. The MS-DOS files are not actually placed on the disk. Use SYS to put the operating system on the disk.

/4

formats a 360K double-sided diskette in a high-capacity (1.2 MB) disk drive.

Do not use FORMAT with an ASSIGN, a JOIN, or a SUBST in effect or with network drives.

Go To Label (GOTO)

GOTO is an internal command for batch files. Batch processing continues with the command on the line below the specified label. GOTO is issued as a separate command except when used with the IF command. IF uses GOTO on the same command line as IF.

label

is a string of up to eight alphanumeric characters. The label in the GOTO command must match a label in the batch file. The label in the batch file must be preceded by a colon. A colon starting a command line tells MS-DOS to ignore the line.

The line containing the label can precede or follow the GOTO command in the batch file. When the label precedes the GOTO, the batch file goes into a loop. When the label follows the GOTO, batch processing skips the intervening commands. If the batch file contains a GOTO command without a label, the batch file ends.

6

In the following example, the double = tells MS-DOS that this is an "equal to" command, not an assignment command, such as SET PATH = A:\COMDIR.

```
copy con private.bat↵
echo off↵
if %1 = = security goto greet↵
echo Access denied↵
goto end↵
:greet↵
echo Welcome! Enter your name!↵
:end↵
^Z↵
```

To process PRIVATE.BAT and to substitute the word "security" for the replaceable parameter %1, type

private security↵

MS-DOS displays

```
A>echo off
Welcome! Enter your name!
```

If you specify any word other than "security," MS-DOS displays

```
A>echo off
Access denied
```

ECHO OFF stops the display of batch commands during batch file processing. MS-DOS replaces %1 with the name you entered when you specified the batch file. MS-DOS compares the name to the one in the IF command. If there is a match, the welcome and the prompt for a name are displayed. If there is no match, the next command processed is the "Access denied" message. MS-DOS then goes to the command following the label :END, which is the end-of-file marker (^Z). The labels are not displayed during batch file processing because MS-DOS ignores lines beginning with a colon. You can add nondisplayed comments to a batch file by starting each comment line with a colon.

Load Graphics Table (GRAFTABL)

GRAFTABL↵

GRAFTABL is an external command that loads a table of additional character data for the color/graphics adapter into memory. Use GRAFTABL with a color/graphics adapter set to graphics mode. It loads a table of data into memory that allows ASCII characters 128 through 255 to be displayed. You need to load GRAFTABL only once after you load MS-DOS. It remains resident until you reset the system.

Print Graphics Display Screen (GRAPHICS)

GRAPHICS ⏎

GRAPHICS is an external command that prints the contents of a graphics display screen on printers that support the printing of individual dots.

Enter the GRAPHICS command after loading MS-DOS. When you want a printout of the screen, press Shift-PrtSc.

If the screen is in text mode, the printout will be in text mode.

If the screen is in the 320×200 color graphics mode, each screen dot is printed in one of four shades of gray.

If the screen is in the 640×200 color graphics mode, the screen is printed sideways on the paper.

Printing speed is dependent on the speed of the printer being used.

Display Command History (HISTORY)

HISTORY [n] ⏎

HISTORY is an internal command that records the commands you have entered. You can display previous commands, and you can select a command from the history to re-enter into the MS-DOS command-line template. Then, using the function keys as described in Chapter 5, you can edit and re-enter that command without retyping the entire line.

HISTORY can store up to 256 characters. HISTORY commands themselves are not stored in the command history. You can abbreviate HISTORY to HI.

Entering HISTORY with no parameters displays the command history onscreen, like this:

```
1.    <most recent command line in HISTORY>
 .        .        .        .       .
 .        .        .        .       .
n.        .        .        .       .
```

Commands too long to fit the display line wrap to the next line.

n

is the number of a command stored in the history. The **n**th command from the history is placed into the command-line template.

For example, enter

hi 5(cr)

MS-DOS puts the fifth line from the command history into the command-line template. You can execute the command without changing it by pressing function key F3, or you can edit the command as described in Chapter 5.

If Condition (IF–EXIST)

IF [NOT] string1 == string2 command⏎

or

IF [NOT] EXIST [path]filename.ext command⏎

or

IF [NOT] ERRORLEVEL n command⏎

IF is an internal command for batch files. If a condition is true, MS-DOS processes the command specified in the IF command line. Otherwise, MS-DOS processes the next command in the batch file.

NOT

processes the command when the IF NOT condition is true.

string1 == string2

compares one alphanumeric string of characters (string1) with another string of alphanumeric characters (string2). If string1 matches string2, the command is processed. To MS-DOS, == means "equal to." String1 is usually a replaceable parameter, such as %1. You cannot use command delimiters, such as commas, within the string parameters. Uppercase and lowercase letters must match.

For example, this batch file uses the IF command and the string parameters to verify system users:

```
copy con users.bat⏎
echo off⏎
if %1 == George goto label⏎
if %1 == Jack goto label⏎
if %1 == Mary goto label⏎
echo You must be an approved user to operate the system.⏎
goto end
:label⏎
echo Welcome! Enter your command.⏎
:end
^Z⏎
```

When you specify USERS and a name to be processed, MS-DOS replaces the %1 parameter in the first IF command with the specified name. MS-DOS compares that name (string1) to the string of characters in string2. If they are equal, MS-DOS displays

```
Welcome! Enter your command.
```

If there is not a match in the first command, the next command is processed. If MS-DOS processes the three IF commands in the file without a match, MS-DOS displays

```
You must be an approved user to operate the system.
```

EXIST [path]filename.ext

tells MS-DOS to look for the specified file in the specified path. If you do not include a path, MS-DOS searches the current directory on the default drive. If the file is found, the command is processed. For example, this command in a batch file searches for the file JOBS in the MGR subdirectory on the diskette in the default drive:

if not exist \mgr\jobs echo Can't find \mgr\jobs.↵

If JOBS is found, MS-DOS does not process the ECHO command. If \MGR\JOBS does not exist, MS-DOS displays

```
Can't find \mgr\jobs.
```

ERRORLEVEL n

tests for a program failure. This parameter tells MS-DOS to check for an error level number where **n** is the specified number. The next command is processed if the previous program run by MS-DOS had an exit code the same as **n** or higher.

command

can be any of the MS-DOS commands. The most common command used with IF is GOTO.

Join Drives (JOIN)

JOIN [d1:] [d2:path]⏎

or

JOIN [d1:] [/D]⏎

JOIN is an external command that joins a specified drive to a directory or another drive to create one directory structure from two separate directories. The JOIN command also displays current joins or disables a join.

d1:

specifies the drive to be joined to a directory. The entire directory structure of the drive, beginning with the root, is joined with the specified directory, regardless of the current directory.

d2:path

specifies the directory that the drive will be joined to. The directory must be one level below the root. The JOIN command creates the directory if it does not already exist. The directory must be empty in order for the join to succeed. (An empty directory contains only the . and .. entries.)

The message "Directory not empty" is displayed if you try to join a drive to a directory that contains files. The path you specify cannot be the current directory. The message "Invalid parameter" is displayed if you try to join a drive with the current directory.

/D

disables the join for the specified drive.

Type the JOIN command without parameters to display the drives and directories that are currently joined.

You cannot refer to a joined drive in a command line; you must refer to the directory that the drive is joined to. The message "Invalid drive specifier" is displayed if you use a joined drive as a drive specification.

You cannot use a network drive as a parameter of the JOIN command. The message "Cannot JOIN a network drive" is displayed if you specify a network drive.

Do not use JOIN with a drive that is part of a SUBST or an ASSIGN. Do not use BACKUP, FORMAT, DISKCOPY, DISKCOMP, or RESTORE while a join is in effect.

6

Load Keyboard (KEYBxx)

KEYBFR ↵

KEYBGR ↵

KEYBSV ↵

KEYBUK ↵

KEYBxx is an external command that loads one of four foreign-keyboard programs into memory. The program you load replaces the American keyboard program that is resident in ROM BIOS. Each of the four foreign-keyboard programs increases the size of MS-DOS in resident memory by a different amount, although each is approximately 2 KB.

KEYBFR

loads the French keyboard.

KEYBGR

loads the German keyboard.

KEYBSV

loads the Swedish keyboard.

KEYBUK

loads the British keyboard.

You can alternate between the American keyboard program and the foreign-keyboard program once the latter has been loaded with the KEYBxx command. While you are in the foreign-keyboard program, simultaneously press Ctrl-Alt-F1 to switch to the American keyboard program. While in the American program, simultaneously press Ctrl-Alt-F2 to switch to the foreign program.

You should load only one foreign-keyboard program into memory after you load MS-DOS. The program remains in memory until you reset the operating system. If you do load a second program, the first remains in memory but you cannot return to it.

You can put the KEYBxx program in an AUTOEXEC.BAT file or a CONFIG.BAT file if you want to load the keyboard when you load MS-DOS. The SELECT program creates an AUTOEXEC.BAT file that loads one of the keyboard programs.

You can create accented characters by using the "dead keys," which are keys that have no effect unless they are used with another key. To create an accented character, press and release the accent key. Then, press and release the key for the character that is to be accented.

Press the accent character and then press the Spacebar to enter the accent key itself.

The possible dead key combinations are as follows:

French   ä Ä ë ï ö Ö ü Ü ÿ â é î ô û

German   á é É í ó ú à è ì ò ù

Swedish   à è ì ò ù é É á í ó ú ä Ä ë ï ö Ö ü Ü ÿ â ê î ô û

British   (no dead keys are supported)

Volume Label (LABEL)

LABEL [d:][volume label]⏎

LABEL is an external command that creates, changes, or deletes a volume label on a diskette or fixed disk volume.

d:

is the drive letter of the diskette or volume you want to label. If you omit the drive specifier, the default drive is assumed.

volume label

specifies the volume label, which can be up to eleven characters. You can use all valid filename characters except a period. If you omit the volume label parameter, the LABEL command displays the current volume label and prompts you to enter a label.

To delete a volume label, type the LABEL command without the volume label parameter. Then, press the Enter key after the prompt is displayed. The following prompt appears:

```
Delete current volume label (Y/N)?
```

Type Y and press the Enter key. The volume label is deleted.

You should not use the LABEL command with network drives or drives that are part of a SUBST or an ASSIGN.

List Files (LS)

LS [options] [files or directories]⏎

LS is an external command that produces a sorted listing in various directory formats of a file or list of files. In the LS listing, names of subdirectories are followed by a \ (such as MGR1\).

options can be any of the following switches:

/B

sorts backwards (Z to A).

/E

sorts output by file extension.

/G

lists files "toGether." Files are not separated by directory.

/L

lists the directory entries in long format. The filename, file attributes, date, time, and size are listed for each file. Long-format listings are not sorted.

/R

recursively lists all files in subdirectories of any directory specified. You can use this command switch to list the contents of all directories on a drive. LS /R lists each subdirectory followed by all the files in that directory.

/T

sorts output by date and time of creation, rather than alphabetically.

/1

lists only one filename per line.

files or directories

is a list of one or more filenames and/or directories. If this parameter is omitted, LS lists the current directory.

The following example alphabetically lists the files and subdirectories in the current directory:

ls ↵

The next example lists, in long format, the files in the TEMP directory on drive B with .ASM extension:

ls /l b:\temp*.asm ↵

You will see a listing on your screen similar to this:

```
a----w      1430    24-SEP-85    12:57 BLDTYPE.ASM
a----w       616    20-MAR-86    10:48 BRITE.ASM
a----w      6247    02-FEB-86    12:37 DTIME.ASM
a----w     60491    10-OCT-85    10:48 HMANAGER.ASM
a----w       229    09-FEB-86    17:44 OMANAGER.ASM
5 files.
```

Make Directory (MKDIR)

MKDIR [path]⏎

MKDIR is an internal command that makes a new subdirectory beneath the root directory, which MS-DOS created when you formatted your disk. You can abbreviate MKDIR to MD. See Chapter 3 for more information on creating and using subdirectories.

path
is the path through the subdirectory hierarchy to the subdirectory you want to create. The last directory name in the path is the name of the new subdirectory.

MS-DOS can create a subdirectory only if a directory above it exists. An initial \ always indicates the root directory. An initial directory name without a \ indicates a subdirectory below the current directory.

For example, if MS-DOS is at the root directory and you enter

mkdir mgr1.nam⏎
mkdir mgr2.nam⏎
mkdir mgr3.nam⏎

MS-DOS creates three subdirectories at the first level beneath the root directory. Each subdirectory is listed in the root directory. You can verify this by entering the DIR command. MS-DOS displays

```
Volume in drive A is JUNE 8 1986
Directory of A:\

MGR1     NAM <DIR>      6-30-86    9:00a
MGR2     NAM <DIR>      6-30-86    9:00a
MGR3     NAM <DIR>      6-30-86    9:01a
       3 File(s)    370683 bytes free
```

"Directory of A:\" indicates the root directory. Subdirectories are displayed with < DIR > ; files are not displayed with this attribute.

Once a directory exists, you can make a subdirectory below it. For example, this command creates the subdirectory JOBS below the subdirectory MGR1.NAM:

mkdir mgr1.nam\jobs ↵

You can enter a directory name without a \ in a MKDIR command to add a subdirectory to the directory you are currently using. For example, if you are working in the MGR1.NAM directory, instead of issuing the preceding command, you can specify

mkdir jobs ↵

Once the JOBS subdirectory has been created, you can create a subdirectory below it:

mkdir mgr1.nam\jobs\benefits ↵

or, if the current directory is JOBS, enter

mkdir benefits ↵

You can make as many subdirectories as you want depending on the amount of space on your disk. Subdirectory names can be 1 to 8 characters followed by an extension of 1 to 3 characters. The valid characters for filenames are valid for directory names. You can duplicate file or directory names as long as they are in separate directories. When you are creating subdirectories, do not use more than 63 characters in the command, including the \.

When you finish making your directory hierarchy, it is a good idea to enter the CHDIR \ command. This command returns you to the root directory. You should always return to the root directory so that you do not accidentally create an entry in the wrong subdirectory.

Set Operation Modes (MODE)

MODE is an external command that sets operation modes for the video display and for printer or serial ports. There are four options for this command. Each has its own function.

MODE LPT#:[n][,[m][,P]] ↲

Sets mode parameters for a printer

MODE n ↲

Sets mode parameters for a video display

MODE COMn:baud[,parity[,databits[,stopbits[,P]]]] ↲

Sets mode parameters for a serial port

MODE LPT#: = COMn ↲

Redirects parallel printer output to a serial port

Set Mode Parameters for a Printer

Use this option to set the operation mode for a printer.

MODE LPT#:[n][,[m][,P]] ↲

#	Establishes the printer number (1, 2, or 3).
n	Establishes the number of characters per line (80 or 132).
m	Establishes the number of vertical lines per inch (6 or 8).
[,P]	Establishes continuous retry on timeout errors.

If you do not specify a value for **m** or **n**, or if the value you specify is invalid, then the mode for that operation does not change. You can stop continuous retries by pressing Ctrl-Break. If you do not want timeout errors to continuously retry, do not specify the P parameter.

For example, if you want to set line printer number 2 for continuous retry and 132 columns with 8-line-per-inch spacing, enter

mode lpt2:132,8,p ↵

Set Mode Parameters for a Video Display

This option sets the operation mode of the video display(s).

MODE n ↵

n   Establishes the type of display. If the color dip switch is set, specifying MONO does nothing to the display. You must set the dip switch to change the type of display from monochrome to color. The settings are as follows:

40 = 40 character display width (color/graphics display mode)
80 = 80 character display width (color/graphics display mode)
BW40 = Black and white, 40 character display width
BW80 = Black and white, 80 character display width
CO40 = Color, 40 character display width
CO80 = Color, 80 character display width
CO320 = Color/Graphics, 320 × 200 display
BW320 = Color/Graphics, 320 × 200 display
BW640 = Color/Graphics, 640 × 200 display
MONO = Monochrome, 80 character display width

Set Mode Parameters for Serial Communications Ports

This option allows you to set the operation mode for serial ports.

MODE COMn:baud[,parity[,databits[,stopbits[,P]]]] ↵

n　　　　Establishes the serial port number (1 or 2).

baud　　Establishes the baud rate (110, 150, 300, 600, 1200, 2400, 4800, or 9600). Only the first two digits are required (for example, 11, 15, 30, 60, and so on).

parity　Establishes the parity mode (N = none, O = odd, E = even). The default is E (even).

databits　Establishes the number of bits in a serial byte (7 or 8). The default is 7.

stopbits　Establishes the number of bits signaling the end of a serial byte (1 or 2). If the baud rate is 110, the default is 2; otherwise, it is 1.

[,P]　　Causes continuous retries to occur on timeouts. This option is useful when you use a serial printer that cannot keep up with the serial rate. Press Ctrl-Break to stop the retry loop. If you do not want the timeout errors to continuously retry, reinitialize the serial port with the P parameter.

You must specify values for the serial port number and the baud rate; if you omit values for the other parameters, MODE assigns the default values. When you do not enter a parameter, enter a comma for the value omitted.

For example, to set up serial port number 2 for 9600 baud, even parity, 8 databits, 2 stopbits, and continuous retries, enter

mode com2:96,e,8,2,p ↵

To use the default values for this command, enter

mode com1:12,,,,p ↵

Redirect Parallel Printer Output to a Serial Port

This option redirects output for a parallel printer to an asynchronous communications adapter. You can redirect output for the logical printers to the serial I/O ports. Before you use this option, you must use the preceding option to initialize the serial port.

MODE LPT#: = COMn ↵

\#   Establishes the printer number (1, 2, or 3).

n   Establishes the serial port number.

The following command causes all outputs to line printer number 3 to be redirected to serial port number 2:

mode lpt3: = com2 ↵

To disable the redirection for the printer number 3, enter

mode lpt3: ↵

More Screen (MORE)

MORE is an external filter command that passes through the data in a
file and displays one screen of data. At the bottom of the screen, this
message is displayed:

```
-- More --
```

Press Enter or the Spacebar to display another screen of data. The
MORE message displays with each new screenful of data until MS-
DOS reads the entire file.

MORE is used mostly with other commands to pipe data (see Section
5.3). For example, you can display a directory one screen at a time:

dir | more ↵

The following command uses MORE in a pipe and filter operation:

dir | find "DIR" | more ↵

The current directory is piped as input to the FIND command. FIND
searches the directory for all DIR entries. The output of FIND is piped
as input to MORE. The MORE command stops the directory display
when the screen is full. When you press the Enter key, another screen
of directory entries is displayed.

For more information on filtering data, see Section 5.3.

Move or Rename Files (MV)

MV [path]filename.ext [path]filename.ext␘

or

MV [path]filename.ext... path␘

MV is an external command that moves files. If you name two files in an MV command, the first file overwrites the second file, and the original file is deleted. If you name a file and a directory, the file is copied into the directory, and the original file is deleted. If you name multiple files and a directory, all the files are moved into the last named directory.

path

is the path through the subdirectory system to the file(s) you want to move, or to the file or directory where you want to move the file(s).

filename.ext

In the first parameter, **filename.ext** is the name of the file(s) you want to move. If the second parameter names a file, the first file is copied into the second file, and the original file is deleted. If the two files are on the same drive, MV renames the first file as the second filename, and deletes the second file if it already exists.

You can use wildcard characters in the first filename if you are moving files into a directory. If you use wildcard characters in the first filename of an MV command moving files into another file, however, MV moves only the first matching file into the destination file. Then MV displays an error message like this:

```
I already moved d:FILENAME.1 to d:DESTFILE
and will not now move d:FILENAME.3 to d:DESTFILE
```

The following example moves all .EXE files to the BIN directory on drive B:

```
A>mv *.exe b:\bin↵

    mv MV.EXE          to (directory) b:\bin
    mv CONCAT.EXE      to (directory) b:\bin
    mv FORMAT.EXE      to (directory) b:\bin
    .   .  .  .  .  .  .  .  .  .  .  .  .  .  .
    .   .  .  .  .  .  .  .  .  .  .  .  .  .  .
```

This MV command renames MEMO.MEM to SEPT28.MEM:

mv memo.mem sept28.mem↵

You can refer to any number of drives in an MV command. For example, this MV command moves all .EXE files and all files with names starting with SAVE on drive A to the BACKUP directory on drive B:

mv a:*.exe a:save*.* b:\backup↵

6-86

Path to Command File (PATH)

PATH [path][;path]...⏎

PATH is an internal command that searches first the current directory then the other specified directories for command or batch files (.COM, .EXE, or .BAT files **only**). PATH allows you to load external commands and batch files from a subdirectory without typing the full path to the subdirectory each time.

PATH without parameters displays the previously specified PATH command. For example, if you enter

path \cmd⏎

and then enter

path⏎

MS-DOS displays

```
PATH=\CMD
```

If a PATH command was not previously specified, MS-DOS displays the message "No Path."

;

used with PATH and no other parameters tells MS-DOS to search only the current directory for a command or batch file. A semicolon between directory names in the path specifies more than one path (indicated by the \) for the command or batch file search.

path

is the path through the subdirectory system to the subdirectory containing the external commands and batch files.

Suppose BIN is the first-level subdirectory and contains all the external commands. You can specify

path \bin ↵

Then you can enter any external command without specifying the BIN subdirectory, regardless of the directory you are currently using.

If you want MS-DOS to search more than one drive or directory for a command or batch file, you can specify

path a:\bin;a:\bin\utils;c:\pascal\programs;d: ↵

MS-DOS searches the current directory first, and then the PATH directories in the order that they were specified. In the preceding example, the directory A:\BIN is searched before A:\BIN\UTILS. The first .COM, .EXE, or .BAT file found whose name matches the command entered will be loaded.

If your PATH command contains an invalid delimiter or parameter, such as the wrong drive name, MS-DOS does not detect it until it looks for a command or batch file. Then MS-DOS displays either or both of the following error messages:

```
Invalid drive in search path

Bad command or file name
```

Pause Batch Process (PAUSE)

PAUSE [comment]⏎

PAUSE is an internal command for batch files that suspends batch processing and displays: "Strike any key when ready...". Pressing any key except Ctrl-C resumes processing the batch file. If you press Ctrl-C, MS-DOS displays

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

If you press Y, the remainder of the batch file is not processed, and the MS-DOS command prompt is redisplayed.

At each PAUSE in the file, MS-DOS stops. You can decide if you want to end processing or do some other action, such as changing diskettes.

comment
is any string of characters (121 maximum) that you specify. The characters are displayed with the "Strike any key when ready..." message.

For example, this command stops batch file processing:

pause Insert a diskette in drive b⏎

MS-DOS displays

```
Insert a diskette in drive b
Strike any key when ready...
```

Print Queued Files (PRINT)

PRINT [d:]filename.ext [/C] [/P] [/T] ... ↵

PRINT is an external command that puts files in a print queue and prints while MS-DOS is doing other tasks. Only files from the current directory can be put in the print queue.

When you specify PRINT for the first time after MS-DOS starts, MS-DOS displays this message:

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

The list device is the printer. Specify the valid name for your list device, such as COM1, COM2, or LPT. If you press the Enter key, MS-DOS uses PRN as the list device. PRN is a logical device name equivalent to LST.

After you give the device name for your printer, MS-DOS displays the message "Resident part of PRINT installed." If you give an invalid assignment name for the device, MS-DOS displays "List output is not assigned to a device." Be sure that the list device is attached to your computer.

The list device cannot be used when PRINT is being processed. MS-DOS displays an error message until the files are all printed or printing is terminated with the /T parameter.

d:
is the name of the drive containing the directory of files that you are currently using. If you do not specify a drive, MS-DOS uses the default drive. Do not remove the diskette from the drive while the files are being printed. If MS-DOS tries to access the file and the drive is not ready, MS-DOS displays the message "Drive not ready."

If MS-DOS finds a disk error, the file being printed is canceled. The printer types a disk error message, advances to the next page, and the printer alarm sounds. MS-DOS then prints the rest of the files in the print queue.

filename.ext

is the name of the file(s) that you want to print. MS-DOS queues and prints the files in the order you specify in the command. You can specify a maximum of 10 files. Entering PRINT and the filename(s) without any other parameters queues and prints the files:

print chapt1.doc chapt2.doc chapt3.doc ↵

MS-DOS displays the name of the drive and the name of the first file being printed:

```
A:CHAPT1      .DOC is currently being printed
```

The names of the remaining files in the queue are also displayed:

```
CHAPT2      .DOC is in queue
CHAPT3      .DOC is in queue
```

MS-DOS prints a file, moves to the next page, then prints the next file. Tabs are expanded with blanks to the next eighth column.

Wildcard characters can be used in the filename and its extension (see Section 2.3).

/C

cancels the named file(s). The filename preceding the /C and subsequent filenames are canceled until MS-DOS finds a /P in the command line.

For example, the following command cancels CHAPT1.DOC, CHAPT2.DOC, and CHAPT3.DOC from the print queue and prints CHAPT4.DOC:

print chapt1.doc /c chapt2.doc chapt3.doc chapt4.doc /p↵

This command removes all CHAPT?.??? files from the print queue:

print chapt?.* /c↵

If there are no other files in the print queue and a file is printing—for example, MORE.TXT—and you enter

print more.txt /c↵

MS-DOS prints

```
A:MORE     .TXT Canceled by operator
```

The printer advances one page, and the printer alarm sounds once. MS-DOS then displays "PRINT queue is empty."

If you specify

print more.txt echo.txt↵

MS-DOS displays

```
A:MORE     .TXT is currently being printed
ECHO       .TXT is in queue
```

If you then type

print more.txt /c↲

while MORE.TXT is printing, MS-DOS does the following:

▶ Stops printing MORE.TXT

▶ Prints the operator-canceled message

▶ Advances the paper one page

▶ Sounds the printer alarm

▶ Prints the next file in the queue

▶ Displays

```
A:ECHO    .TXT is currently being printed
```

/P

prints the named file(s). The file preceding the /P and subsequent files are printed until MS-DOS finds a /C in the command line. For example, this command prints CHAPT5.DOC and CHAPT6.DOC, cancels CHAPT7.DOC, and prints CHAPT8.DOC:

print chapt5.doc /p chapt6.doc chapt7.doc /c chapt8.doc /p↲

/P is the default if you specify only PRINT and the filename.

/T

deletes all queued files from the print queue. When you use /T, do not specify any other parameters. For example, this command empties the print queue:

print /t↲

MS-DOS displays the message "PRINT queue is empty" and prints "All files canceled by operator."

Change MS-DOS Prompt (PROMPT)

PROMPT [prompt-text]↵

PROMPT is an internal command that changes the MS-DOS default prompt. PROMPT without parameters designates the default prompt to be A > .

prompt-text

represents alphanumeric character(s) that you specify.

For example, you can specify the prompt as COMMAND by typing

prompt COMMAND↵

MS-DOS displays the new system prompt instead of the A > prompt:

```
COMMAND
```

If you type the following, the A > prompt returns:

```
COMMANDprompt↵
A>_
```

The next example changes the prompt from A > to READY, and then reinstates the A > prompt:

```
A>prompt READY↵
READYprompt↵
A>_
```

You can also insert special characters in the prompt by specifying $c for the prompt-text, where c is:

$ The $ character

t   The current time

d   The current date

p   The name of the current directory of the default drive

v   The MS-DOS version number

n   The default drive

g   The > character

l   The < character

b   The | character

s   A leading space only

e   The Esc character

_   Carriage return/line feed, which returns to the beginning of the next line on the screen

The special characters inserted with $ must be lowercase. If you specify $ and any other character than the ones described, MS-DOS ignores the command.

To use the current time and date as a system prompt, type

prompt TIME = t_DATE = $d ↵

MS-DOS displays, as a two-line command prompt, the current time and date:

```
TIME=10:54:30.50
DATE=Mon 7-04-86
```

Recover Files (RECOVER)

RECOVER [path]filename.ext⏎

or

RECOVER d:⏎

RECOVER is an external command. Use this command when MS-DOS cannot read a file because of a defective disk. If a file or a diskette directory has a bad sector, this command recovers the file or the entire disk. If there is a bad sector in the directory, this command recovers all the files on the disk.

CAUTION: Use RECOVER only when MS-DOS cannot read a file because of a defective disk. Running RECOVER on a good disk results in all the files being renamed.

path

is the path through the subdirectory system to the file you want to recover.

filename.ext

is the file that has the bad sector. The file is read sector by sector from the disk. MS-DOS marks the bad sector, skips it, and allocates it to a system table. The data in the bad sector is not recovered. MS-DOS does not make any more file allocations to that sector.

d:

is the name of the drive you want to recover. Entering RECOVER and the drive name with no parameters recovers all the files on the disk. MS-DOS scans the File Allocation Table (FAT) for chains of allocation units (the amount of space used on a disk for a file). MS-DOS creates a new root directory. Each entry in the root directory is renamed as:

FILEnnnn.REC

nnnn is a sequential number from 0001. Each FILEnnnn.REC points to the beginning of one of the allocation chains.

If the root directory cannot contain all of the allocation unit chains, RECOVER displays a message. The unrecovered chains are left in the File Allocation Table (FAT). Reissue RECOVER and the drive name when you have made more room in the root directory.

Remark (REM)

REM comment↵

REM is an internal command for batch files. During batch processing, each REM command displays the comment stored on that command line in the batch file.

comment
is a string of characters (123 maximum) that you want displayed when the REM command is processed in the batch file. The only delimiters you can use in the comment are a space, tab, or comma. Here is an example:

```
REM This is a file to display the main directory
PAUSE Insert disk in drive B:
DIR B:
```

You can use a period (.) in place of REM to indicate a comment line. For the first line in the example, you can enter

```
.This is a file to display the main directory
```

Rename Files (REN)

REN [path]filename.ext filename.ext↵

REN is an internal command that renames files.

path

is the path through the subdirectory system to the directory containing the file you want to rename. The path always includes the drive name if the drive is not the default drive.

filename.ext

The first filename is the name of the file to be renamed. The second filename is the new name. The file itself remains on the diskette where it was originally created.

You can use wildcard characters in either filename. This command changes all the files with the .DAT extension to the .CMD extension:

ren *.dat *.cmd↵

The following command renames the file PHNLIST on drive B to PHNNUM. The file remains on drive B.

ren b:phnlist phnnum↵

If you rename a file with a name that is already in the directory, MS-DOS displays the message "Duplicate file name or File not found."

Restore Files (RESTORE)

RESTORE d: [d:][path][filename[.ext]] [/S] [/P] ↵

RESTORE copies files from a backup diskette to the fixed disk. You must use RESTORE before you can work with any of the backup files.

/P

prompts you to confirm whether to overwrite files marked "read-only" or those modified since the backup.

/S

restores all files in all levels of subdirectories within the specified directory. If you used this switch to back up the files, you must also use it to restore them.

If you do not specify a pathname, the current directory is restored. You can use wildcard characters to restore particular files or groups of files. For example, this command restores all files on drive A that have the filename extension .DOC to the current directory on drive C:

restore a: c:*.doc ↵

This command restores all files (/S) from drive A to drive C:

restore a: c:\ /s ↵

In the preceding example, MS-DOS prompts you to insert diskettes into drive A as it restores all files from the series of backup diskettes. The directories and filenames are displayed as files are restored. The system alerts you with a message if disk space runs out before all the backup files are restored.

The following command restores all files from drive A to drive D, and prompts you (/P) when a file is encountered that was modified since the last backup:

restore a: d: /p ↵

MS-DOS asks you whether to overwrite the file with the backup or to leave the existing version of the file intact.

RESTORE sends an error level number to the batch command IF–ERRORLEVEL. (See the IF command for details on how you can use these error levels to control subsequent error-level processing.) The following error levels are set by RESTORE:

0   Normal completion

1   No files were found to restore

3   Terminated by user (Ctrl-Break or Esc)

4   Terminated due to error

Remove Directory (RMDIR)

RMDIR [path] ↵

RMDIR is an internal command that removes the last named subdirectory in a directory hierarchy. You can abbreviate RMDIR to RD. Chapter 3 gives more information about RMDIR and the directory system.

path
is the path through the directory hierarchy to the subdirectory you want to delete.

RMDIR will not delete a subdirectory if it is not empty. If you try to delete the current directory or a subdirectory containing any entries other than the . and .. (notations indicating this directory level and the next higher level), MS-DOS displays the message "Invalid path, not directory, or directory not empty."

Delete (DEL) all files from the subdirectory you want to remove. Then delete all the directory entries from the subdirectory with the RMDIR command.

You can enter RMDIR and the directory name without a backslash to delete a subdirectory entry in the current directory. Suppose you have this directory hierarchy:

\MGR1.NAM\JOBS\BENEFITS

If you are currently working with the JOBS subdirectory, you can delete BENEFITS from JOBS by entering

rmdir benefits ↵

If you are using any directory except BENEFITS, this command removes the BENEFITS subdirectory, if it is empty:

rmdir \mgr1.nam\jobs\benefits ↵

To remove the JOBS subdirectory, return to the root directory using the CHDIR command. Make sure JOBS is empty, and then enter RMDIR with the path to JOBS.

Search Files (SEARCH)

SEARCH [where: files or directories] [which: matching constraints]
[what: actions]⏎

SEARCH is an external command that has these three functions:

▶ Searches through the specified files and/or directories

▶ Selects matching files according to the specified constraints

▶ Either lists the matching file(s) or performs specified action(s) on the matching file(s)

You specify the action you want SEARCH to perform with command switches. With SEARCH you can:

▶ Move, copy, or delete files

▶ Archive files by creating a disk archive file

▶ Change file attributes (such as read-only or read/write status)

▶ Find the sum of file sizes

▶ Delete empty subdirectories

Additional switches allow you to request prompting, exclude subdirectories from the search, and copy entire subdirectory hierarchies.

Examples of SEARCH

Before describing the specific options of SEARCH, this section gives some common usages.

search /name *.pas!*.asm /cp b:\files⏎

Copies all files ending in .PAS or .ASM into the FILES directory on drive B.

search f:my\stuff /mv a: /t /rmdir⏎

Moves all of the files in F:MY\STUFF to drive A, preserving the subdirectory structure within MY\STUFF. For example, the file MY\STUFF\X\Y\Z is copied to A:\MY\STUFF\X\Y\Z. Any empty directories within MY\STUFF are removed, so that after the move, the subdirectory F:MY\STUFF is completely erased.

search . /backup b:\archive /nr⏎

Backs up all files in the current directory (.) that have been modified since the last backup to directory B:\ARCHIVE. If you include /T in the command, the structure of the current directory's subdirectories is preserved on B:\ARCHIVE. The preceding command is an abbreviation for the command:

search . /a /cp b:\archive /attr- a /nr⏎

which finds all files with the archive bit on (/A) and resets the archive bit (/ATTR- A) after copying. Because MS-DOS sets the archive bit whenever a file is written to, this command is an effective way to back up only modified files.

cd old⏎
search . /mv \new /t /rmdir⏎

Moves files from the OLD directory to the NEW directory, preserving the directory structure (/T). Removes any empty directories (/RMDIR) in OLD. This procedure is the easiest way to rename a directory.

search f:\ a:\ /ls /nr⏎

Finds all files on drives F and A, lists them in "long" format (including attribute bits, file size, file date, and name). Does not traverse (/NR) the subdirectories of drives F and A.

search c:\ /tarc b:\c.tar ↵

Copies all of C:\ to the TAR file B:\C.TAR. A TAR (tape archive) file is a single file that contains all the copied files. It is faster to create this file than to use the COPY command to copy individual files. A selective RESTORE is available if you want to extract an individual file or files from a TAR file.

TAR files provide the fastest way to copy a collection of files from one computer to another while preserving the directory structure and the time and date of the copied files.

search c:\ /autotarc b:\ ↵

Copies all of C:\ to successive diskettes inserted in drive B. The only difference between this command and the preceding command is that SEARCH creates and names as many TAR files as needed: 1.TAR, 2.TAR, 3.TAR, and so on. With the previous command, if all of C:\ does not fit on B:\C.TAR, SEARCH asks you whether or not to continue. With /AUTOTARC, SEARCH automatically continues and prompts you for new diskettes. This procedure is an easy way to back up an entire 10-MB disk volume to diskettes.

search @list /backup b:\ ↵

Backs up all files listed in the file LIST to B:\. This command is useful if you have many files that you back up frequently. You can place all the filenames in a file (in this example, LIST). Then you can reference the files with the SEARCH command, preceding the filename LIST with an @.

search . /name *junk* /rm ↵

Starting at the current directory, removes all files that have JUNK somewhere in the name, and have no extension. For example, this command removes the following files:

AJUNK
XJUNKY
JUNK

search . /name *junk* /rm /. ↵

Acts the same as the preceding example, except /. enables * to match . (unlike the normal use of the * in MS-DOS). The command removes the files listed in the previous example, as well as these files:

AJUNK.TMP
XJUNKY.PAS
JUNK.JNK

search . /name *jnk* /rm /. ↵

Removes files such as:

XYZ.JNK
JNKX
XJNK.JNK

Syntax of SEARCH

Because SEARCH has several types of parameters, the syntax description for this utility is divided into sections listing and describing each type of parameter: *Where: Files or Directories*, *Which: Matching Constraints*, *What: Actions*, and *Action and Constraint Modifiers*.

The first section presents the syntax for "where: files or directories" and gives examples showing how to use file and directory names with SEARCH. In the other three sections, the command switches for that set of parameters are presented in table form. Finally, examples showing all the SEARCH command parameters are given in the section, *Hints for Using SEARCH*.

Where: Files or Directories

The first set of SEARCH parameters, files or directories, immediately follows the keyword SEARCH in a command. This parameter list consists of one or more subdirectory names and/or filenames that can include the MS-DOS wildcard characters. The list is the domain from which SEARCH selects files to operate on. If only a drive name is given, SEARCH begins at the current working directory of that drive and searches downward through any subdirectories below.

If you want SEARCH always to start at the root directory in the domain, use a \ at the beginning of the path. For example, type

search a:\ b:\ parameters⏎

SEARCH searches downward from the root directories on drives A and B only. Without the \, the search is relative; in other words, the search starts at the current working directory, which might not be the root directory.

You can mix filenames and directories to specify a domain of specific files and the files in some subdirectories. This command considers the file or directory B:\RTC, and any subdirectories below it, and the two TOD files on the default drive when executing the other parameters:

search b:\rtc tod.src tod.asm parameters⏎

Which: Matching Constraints

The second set of (optional) parameters in a SEARCH command is matching constraints (see Table 6-1). These are command switches that further define which files in the specified domain of files and/or directories you want to affect with your SEARCH command. Only files from the domain that meet all the conditions listed as matching constraints are acted on. You can use each constraint switch only once in a command line.

Table 6-1: Matching Constraint Switches for SEARCH

SWITCH	EFFECT
/NAME filenames	Selects files from the domain that match the filenames following /NAME. To list more than one filename, separate them with exclamation marks, like this: **/name rtc.asm!tod.asm!tod.obj** ⏎ Details about using filenames with SEARCH are given following this table.
/^NAME filenames	Selects files from the domain that do NOT match any of the names following /^NAME. See the examples following this table.
/FNAME pathname	Selects files whose pathnames, including subdirectories and filename, match the names following /FNAME. See the examples following this table for a description of what the pathnames can be.
/^FNAME pathname	Selects files whose full pathnames do not match any of the names following /^FNAME. See the examples following this table.
/ATTR attribute expression	Selects files whose attributes make the specified attribute expression true. The expression consists of letters representing one of the five "file attributes" described here. Multiple letters can be separated with & (for AND), ! (for OR), and ^ (for unary NOT). You can also use parentheses for grouping subexpressions.

SWITCH	EFFECT

The five file attribute letters are

A = archive status is set on (i.e., the file has been edited since the last archive)
D = this "file" is a subdirectory
H = this file is a hidden file
R = this file is read-only
S = this file is a system file

To select read/write files (files that are not read-only) with archive status set on, enter

search a:\ /attr a&ˆr⌡

To display the status of the five file attributes for your files, use the LS /L command (or SEARCH with the /LS action switch).

/COMP directoryname
Compares the files in the domain with files of the same name in the specified subdirectory. Selects only those files that have the same name and are equivalent to a file in the specified directory. Any differences between the two files are displayed.

Include the /T switch with /COMP to preserve the directory tree structure of the domain. With /T, the target directory name for the compare is computed as follows:

1. The drive prefix (and any .\ or ..\ prefixes) are deleted from the directory name of the file found (i.e., the drive name is ignored).

2. The remainder of the directory name including the filename from the domain is appended to the directory named after the /COMP to produce the file which is to be compared.

For example, SEARCH A: /COMP TEST /T compares files on drive A with files in the relative path TEST. If the file A:\X\Y exists, SEARCH compares it to TEST\X\Y but not to TEST\Y.

SEARCH A: /COMP TEST without the /T switch compares A:\X\Y to TEST\Y.

/ˆCOMP directoryname
Selects files from the domain if there is not a file of the same name in the specified path OR (if there is a file of the same name) the two files are not equivalent. Any differences between the two files are displayed. The description of /COMP gives more details.

SWITCH	EFFECT
/EXISTS directoryname	Selects files from the domain whose names match filenames in the specified subdirectory. If /T is used, the path is computed as for /COMP.
/^EXISTS directoryname	Selects files from the domain whose names do not match any filenames in the specified path. If /T is included, the path is computed as for /COMP.
/LE filename	Selects files from the domain whose time of creation (date and time) precedes or is the same as that of the named file. Read LE as "less than or equal to."
/LT filename	Selects files whose time of creation (date and time) precedes that of the named file. Read LT as "less than."
/GE filename	Selects files whose time of creation (date and time) follows or is the same as that of the named file. Read GE as "greater than or equal to."
/GT filename	Selects files whose time of creation (date and time) follows that of the named file. Read GT as "greater than."
/LEDATE filename	Selects files whose date of creation (regardless of time) precedes or is the same as that of the named file. Read LE as "less than or equal to."
/LTDATE filename	Selects files whose date of creation (regardless of time) precedes that of the named file. Read LT as "less than."
/GEDATE filename	Selects files whose date of creation (regardless of time) follows or is the same as that of the named file. Read GE as "greater than or equal to."
/GTDATE filename	Selects files whose date of creation (regardless of time) follows that of the named file. Read GT as "greater than."
/SIZE n	Selects files that are at least **n** bytes in size.
/SIZEB n	Selects files that are at least **n** (512-byte) blocks in size.

File and directory names listed with the /NAME, /^NAME, /FNAME, and /^FNAME switches can have several types of wildcard characters, some of which are themselves switches. The wildcards are as follows:

▶ ? matches any single character, or none.

▶ * matches any number of characters anywhere in the text. In most MS-DOS commands, *A* is equivalent to * because the first * pads to the end of the name. With the /NAME switch, however, *A* matches any text that contains an A.

▶ /. makes wildcards also match the file extension separator (the . between the filename and extension). For example, without the /. switch, you must specify *.* to match all files; using * with the /. switch matches all files.

▶ /\ makes wildcards also match the \ in a pathname. For example, A**.PAS matches A\X\Y.PAS and A\Z\W.PAS, but does not match A\X\Y\Z\Q\R\X.PAS. Using the * with the /\ switch also matches this last path.

▶ [ ] provides multiple-choice selection. If you enclose characters in brackets, SEARCH looks for only the characters listed. For example, [AEIOU]*.ASM matches any filename beginning with a vowel and having an extension of .ASM.

For example, these two /NAME or /^NAME phrases are equivalent:

[bv][tc][1k]fd*.*

[bv][tc][1k]FD* /.

Both phrases match any file with B or V as its first character, T or C as its second, 1 or K as its third, and FD as its fourth and fifth characters. The second example uses the /. switch to match the . between the name and extension. You can use uppercase or lowercase.

This example matches any file in the root directory on drive A or in any subdirectory on drive A which does not have the extension .ASM:

search a:\ /name *.asm↵

The next example matches any file in drive A's root directory or any subdirectory on A whose name does not begin with B or V:

search a:\ /^name [bv]*.* ↵

The last example uses both /NAME and /^NAME. The files selected from the domain, which is the root directory of A, must have an extension of .ASM but cannot begin with BT1:

search a:\ /name *.asm /^name BT1*.* ↵

The difference between the /FNAME and /NAME switches is that /NAME performs the matching test only on the filename, and /FNAME performs its test against the full pathname for that file, starting at the drive name and the root directory.

For example, the following command selects filenames beginning with SRC and having no filename extension in the root directory on drive B, or any subdirectory on drive B that begins with SRC:

search b:\ /fname b:\src* ↵

Adding the /\ switch matches any subdirectories below the first one. Adding /. matches any file extension. For example, this command matches files in any path beginning with B on drive A:

search a:\ /fname a:\b* /\ /. ↵

To match drive designators, remember that the format is D:\. For example, this command selects files from the root directory on drive A that begin with T, or any files in subdirectories on A whose directory names begin with T:

search a:\ /fname a:\t* /\ /. ↵

The last example selects all files from drive A that are not in subdirectory A:\TEMP:

search a:\ /^fname a:\temp* /\ /. ↵

6

What: Actions

The third set of parameters for SEARCH are action switches. These options specify the function to be performed on the files that meet the matching constraints. With action switches you can set and reset file attributes; move, copy, delete, and back up files; print directory entries; and find the sum of the file sizes. Table 6-2 describes the switch settings for actions. You can specify actions marked with an asterisk (*) only once in a SEARCH command.

Table 6-2: Action Switches for SEARCH

SWITCH	EFFECT
/ATTR+ attrs	Sets the specified file attributes on. Specify "attrs" as single-character attributes, as for the /ATTR matching constraint (see Table 6-1). Do not use separators between attribute letters. For example, this switch sets archive and read-only status on:
	/attr+ ar↵
/ATTR- attrs	Set the specified attributes off.
/AUTOTARC pathname *	Is the same as /TARC except that the name of the created TAR file is PATHNAME\1.TAR; if the TARred files do not fit, SEARCH prompts for a new diskette (if applicable) and continues TARring on PATHNAME\2.TAR. This process continues until all files are TARred. This is an effective, automatic way to back up a large disk volume onto diskettes. You cannot specify /AUTOTARC with /TARC or /TARA.
/BACKUP directoryname *	Is an abbreviation for:
	/cp directoryname /attr a /attr- a↵
	That is, this switch finds archivable files (those with archive status) and resets archive status off after the copy. To preserve the tree structure of the files you back up, use the /T switch. You cannot use /BACKUP with /CP.

SWITCH	EFFECT
/CP directoryname *	Copies the matching files into the specified directory. This cannot be used with /BACKUP. The /T option can be used as for the /COMP matching constraint switch (see Table 6-1).
/LS	Lists a directory of matching files in long format. This switch displays file attribute settings, file size, the time and date of creation, and the full pathname. If an attribute is not set for a file, / is printed; if it is set, the attribute is printed. For read/write files, however, w is printed instead of the /. See the matching constraint /ATTR in Table 6-1 for a list of the attributes.
/MV directoryname *	Moves the matching files from the domain into the specified subdirectory. The original files are deleted.

If the /T option is specified, the tree structure of the
source file is preserved. For example, this command:

search a: /mv z /t⏎

finds A:\X\Y; the file Z\X\Y would be created rather
than Z\Y (the file created without /T).

The destination directory name is formed from the source
pathname as follows:
1. The drive prefix and any .\ or ..\ is removed from the
   source directory name.
2. The remainder of the source directory name is
   appended to the directory name after the specified
   /MV directory name.

/PRINT	Lists the names of the matching files from the domain. PRINT is the default action for SEARCH.
/RM	Removes (deletes) the matching files. If both /CP and /RM are specified, the copy is done before the removal. If /MV is specified with /RM, /RM has no effect.
/RMDIR	Removes any empty directories encountered. If /MV is specified, the directory remove is attempted after the files are moved. This command moves all of X to drive B, and then removes all of subdirectory X:

search x /mv b: /rmdir⏎

/SUM	Displays the sum of the sizes of matching files. This is the exact size of the data in the files and does not consider allocation units' round-off.

SWITCH	EFFECT
/TARA filename.ext *	Appends the matching files to an existing TAR (tape archive) file. Otherwise acts same as /TARC. Only one of /TARA, /TARC, or /AUTOTARC can be specified.
/TARC filename.ext *	Creates a TAR (tape archive) file named FILENAME.EXT containing all matching files.
	For example, this command puts all .PAS files on the default drive into the file PASBKUP.TAR:
	search *.pas /tarc pasbkup.tar ↵
	Only one of /TARA, /TARC, and /AUTOTARC can be specified.
/TARX filename.ext *	Extracts the matching files from the tape archive file FILENAME.EXT. The only action you can perform with this option is to /LS, /PRINT, /SUM, or /CP them. The /RM or /MV switches are ignored.
	This command lists the files in the tape archive file PASBKUP.TAR:
	search /tarx pasbkup.tar /ls ↵
	This command copies the files in the given TAR file into directory XYZ on the default drive:
	search /tarx pasbkup.tar /cp xyz ↵

*These switches can appear only once in a SEARCH command.

Action and Constraint Modifiers

The switches listed in Table 6-3 can modify both the matching constraints and the actions listed in a SEARCH command. Single-letter options can be grouped after a single switch signal (such as /AV) if they do not form one of the multiple-letter options. Each multiple-letter option must have a separate signal. For example, you cannot combine the /AV switch with the /NR switch to make /NRAV.

Table 6-3: SEARCH Action and Constraint Modifiers

SWITCH	EFFECT
/A	Matches only files with the archive status on. Short for /ATTR A.
/N	Takes no actions. SEARCH displays what it would have done. The /N switch is equivalent to using /P and always responding negatively.
/NR	Does no recursive searching of subdirectories. SEARCH normally searches any directories you specify and all subdirectories. Using this switch limits searches to only the directories specified.
/P	Prompts (asks yes/no) for /CP, /MV, /RM, /RMDIR, /TARA, /TARC, /TARX, /BACKUP, /ATTR-, or /ATTR + options. Use this switch to verify SEARCH's actions before taking any action that would have a permanent or destructive effect.
/S	Is silent (displays nothing).
/T	Preserves tree structure of directories for /MV, /CP, /COMP, /^COMP, /EXISTS, /^EXISTS, /TARX, and /BACKUP options.
/V	Preserves volume/drive structure. An option for /TARX only, this switch returns files from tape archive to their original volume or drive.
/.	Allows the file extension separator (.) to be wildcarded.
/\	Allows the directory separator (\) to be wildcarded.

Hints for Using SEARCH

▶ Remember that all matching constraints must be met (they are ANDed together) before a file is processed (listed, deleted, moved, or archived) by SEARCH.

▶ You can mix and match any constraints, action switches, and modifying options switches.

▶ If you are doing anything that might cause data to be lost or reformatted, add the /P switch to prompt you for verification of the action, or append the /N switch to display the actions that would occur if you execute the command.

Use the following examples as guidelines when you use SEARCH:

1. If you specify only a drive name in the domain, such as:

   search f: /name *.pas↵

   SEARCH does not necessarily search all the files on that drive. SEARCH starts at the current working directory and skips any higher directories. To search drive F for files with the .PAS extension, either do a CD F:\ before the preceding command, or enter

   search f:\ /name *.pas↵

2. To list all directories on drive D, type

   search d:\ /attr d↵

3. To copy all files that have been modified since the last time you did a backup, enter

   search a:\ /backup b:↵

   This copies modified files to drive B and resets their archive status off. If the archive status of your files is unknown, enter

   search a:\ /attr + a↵

   to set archive status on for all files (so all files will be archived).

4. Because some commands are abbreviations, remember that there are some restrictions. For example, this command

   search a: /backup b: /attr r↵

   is illegal because /BACKUP B: is an abbreviation for /CP B: /ATTR A /ATTR- A, and only one /ATTR is allowed. Instead you should enter

   search a: /cp b: /attr a&r /attr- a↵

5. To find all files created on the same day as SOFTWARE.MEM, on drives A and C, enter

    search a:\ c:\ /ledate software.mem /gedate software.mem ↵

    This command lists all files with dates less than or equal to AND greater than or equal to SOFTWARE.MEM's date. Thus, all files created on that date will be listed.

    To copy all of these files to drive B's working directory, enter

    search a:\ c:\ /ledate software.mem /gedate software.mem /cp b: ↵

6. To list all hidden and system files on drives/volumes A, C, D, and E, enter

    search a:\ c:\ d:\ e:\ /attr slh /ls ↵

7. Before you archive files with SEARCH /TARC, you can see the amount of space that will be occupied by the archived files. If you are about to archive drives/volumes A, C, and D, enter

    search a:\ c:\ d:\ /a /sum ↵

    The results tell you the size of the tape archive file that will be created when these files are archived.

8. To archive the files from drives/volumes A, C, and D, enter

    search a:\ c:\ d:\ /a /tarc b:backup /attr- a ↵

    This command finds all files with archive status on (/A), archives the files into a single file called BACKUP on drive B (/TARC B:BACKUP), and then resets archive status off (ATTR- A).

    If a file being backed up using /TARC or /TARA will exceed the capacity of the diskette, SEARCH prompts you whether or not to continue.

If you respond Y (yes), the backup continues on the same diskette. When the diskette is full, the system displays a message indicating how many bytes of the file were written, and waits for you to insert a new diskette.

If you respond N (no), the TAR process stops. Because archive status is switched off (/ATTR; see Table 6-1) for each file that is backed up, the next TAR will back up only files that you have modified since the last TAR.

9. If you want to re-arrange your directory structure, you can use SEARCH to copy entire subdirectories. For example, to move the subdirectory A:\COMMAND\SOURCES and all of its subdirectory structures, including all files, to C, enter

    search a:\command\sources /mv c:\ /t↵

    In this example, using the /T option preserves the tree structure (\COMMAND\SOURCES\...). Without the /T switch, all the files of all the directories would be copied into the current working directory of C.

10. To remove any empty directories on drives E and F, type

    search e:\ f:\ /rmdir↵

    The /RMDIR switch can be used with any other actions. Only empty directories will be removed. Any drive's current working directory cannot be removed.

11. If there are duplicate files in the directories F:\PROJECTS and G:\MYPROJ, delete the redundant files from G:\MYPROJ with this command:

    search g:\myproj /exists f:\projects /rm↵

    If you want to delete only files that contain the same data as the similarly named files in F:\PROJECTS, enter

    search g:\myproj /comp f:\projects /rm↵

Select Keyboard Layout (SELECT)

SELECT countrycode keyboardcode↵

SELECT is an external command that creates a new MS-DOS diskette that selects a keyboard layout and date and time format for your particular country. SELECT uses DISKCOPY to copy the system diskette with an updated CONFIG.SYS file and an AUTOEXEC.BAT file.

countrycode
is a 3-digit code that determines the date and time format, the currency symbol, and the decimal separator. Some possible codes are as follows:

United States	001
France	033
United Kingdom	044
Sweden	046
Germany	049

keyboardcode
is a 2-letter code that determines the keyboard layout. Some possible keyboard codes are as follows:

France	FR
Germany	GR
Sweden	SV
United Kingdom	UK
United States	US

Set Equivalent Value (SET)

SET [string1 = [string2]] ↵

SET is an internal command that sets one string value equivalent to another string. In subsequent MS-DOS or application program operations, the equivalent value is substituted for the first named string.

String equivalents set by SET are stored in the operating environment. The operating environment is a series of names and parameters built by the command processor and passed to all programs that it invokes. You can SET almost anything into the environment. The only syntax requirement is that you enter a single equals sign (=) in the command line.

Entering SET without any parameters displays all the values that have been set with the SET command. For example, this command

set ↵

lists all the values currently set, such as:

```
path=
prompt=$n$g
load=\utils
```

In the example, path = is the parameter for your path, prompt = is the parameter for your prompt, and load = is the parameter for the path of the last program you invoked.

string1

is an environment-name or variable name that you want to set equivalent to the value of string2. For example, you can duplicate the function of the PROMPT command with SET. Enter this command to change the system prompt to YES?:

    set prompt = YES? ↵

To clear any setting, enter the SET command giving only the first string and an equals sign:

    set prompt = ↵

string2

is the value that you want to substitute for string1 in subsequent operations.

SET is useful if you are doing batch file processing or using an application program that repeats a variable. For example, in a batch file you can create replaceable parameters with names instead of the usual numbers. Replaceable parameters defined as names must have a leading and trailing %, such as %FILE%. You can assign the value to use for that parameter with the SET command:

    set file = domore ↵

When MS-DOS processes the batch file, %FILE% becomes the filename DOMORE. Using this procedure, you do not have to edit a file to change the parameter names before you run the batch file.

Enable File Sharing (SHARE)

SHARE [/F:filespace] [/L:locks]↵

SHARE is an external command that enables file and record sharing calls. These calls prevent data corruption and data loss in a multi-user (network) environment.

/F:filespace

allocates file space in bytes for the area that will store file sharing information. Each open file requires the length of the full filename plus 11 bytes. The default value is 2048 bytes.

/L:locks

allocates space for the number of locks you want. The default value is 20 locks.

If you load SHARE, all reads and writes are checked against the file sharing code. If file sharing is installed and you try to re-install it, the following message is displayed:

```
SHARE already installed
```

Shift Replaceable Parameters (SHIFT)

SHIFT ⏎

SHIFT is an internal command for batch files that moves all replaceable parameters one position to the left:

%1 replaces %0
  ..
  .
%10 replaces %9

When there are more than 10 parameters on a command line, each additional parameter shifts into %9.

For example, this batch file named DIRLOOK.BAT uses the SHIFT command to display as many directories as you enter when you load the batch file:

```
dir /p %1
:nexdir
shift
if %1 = = stop goto lastone
dir /p %1
goto nexdir
:lastone
cd \
```

If you specify

dirlook managers scheds stop ⏎

MS-DOS searches the root directory for the MANAGERS and the SCHEDS directories. During batch processing, MS-DOS displays each batch command (except REM lines or lines beginning with colons) and then performs the command on the line.

MANAGERS is used for %1. Because of the SHIFT command, %1 becomes SCHEDS. PROJ then replaces SCHEDS, and so on. The batch file completes when STOP equals STOP in the IF command.

MS-DOS displays

```
A>dir /p managers
```

Next the MANAGERS directory is displayed.

```
A>shift

A>if scheds == stop goto lastone

A>dir /p scheds
```

Next the SCHEDS directory is displayed.

```
A>goto nexdir

A>shift

A>if stop == stop goto lastone

A>cd \
```

The CD \ at the end of the file returns you to the root directory. See Chapter 3 for information on changing directories, or see CHDIR in this chapter.

Sort Data (SORT)

[command ǀ] SORT [/R] [/ + n] [< [d:]input source]
[> [d:]output source]↵

SORT is an external command that filters data by re-arranging it alphanumerically (using the ASCII collating sequence; see Appendix A). The data to be re-arranged can be the output from an MS-DOS command.

command

is any external MS-DOS command whose output you want to sort. The ǀ symbol tells MS-DOS to direct (pipe) the output from the MS-DOS command to the SORT command, as described in Section 5.4.

For example, this command sorts the current directory alphabetically:

dir ǀ sort↵

The following command uses the output from the DIR command as input to the FIND command. The output of FIND, the names of all the subdirectories in the current directory, is sorted alphabetically and displayed on the screen.

dir ǀ find " < DIR >" ǀ sort↵

/R

tells MS-DOS to reverse the alphanumerical sort (Z to A, or **n** to 0 where **n** is an integer).

6

/ + n

tells MS-DOS to sort the data by columns. **n** is the number of the column where the SORT is to start. For example, this command sorts the directory alphanumerically by filename, which is the first column in the directory display:

dir | sort / + 1⏎

This command does a reverse alphanumeric sort of the directory:

dir | sort /r / + 1⏎

<

tells MS-DOS to use the parameters following the < symbol as input.

d:

is the name of the drive containing either the input data to be sorted or the sorted output data. The drive name is optional when the input or output source is on the default drive.

input source

is the data you want to sort. It can be a disk file or data from a MODEM. For a disk file, specify the name of the file and the path to the file if it is not in the current directory. For a MODEM or any other device, you must specify the physical device name.

>

tells MS-DOS to use the parameters following the > symbol as output. You must include an output source if you include the > symbol. For example, if you enter

dir | sort > ⏎

MS-DOS displays the message "File creation error."

output source

is the destination of the sorted data from the input source. It can be files, printers, or other devices. For files you must include the path to the file unless it is in the current directory. The following command sorts the directory alphabetically and sends the output to the LST device, the printer:

dir | sort > lst⏎

The following command pulls all the records containing the zipcode 95066 from the MAIL.LST file and sorts them alphabetically into the ZIP.COD file:

find "95066" mail.lst | sort > zip.cod⏎

ZIP.COD is created by this command, or if ZIP.COD already exists, its contents are overwritten by the output of SORT. To see the sorted data records in ZIP.COD, enter

type zip.cod⏎

MS-DOS displays

```
---------- mail.lst
```

followed by an alphabetical list of each 95066 record.

Substitute Drive Specifier (SUBST)

SUBST [d1:] [d2:path]↵

   or

SUBST [d1: /D]↵

SUBST substitutes a different drive specifier for a drive or path. SUBST is useful if you are using an application that does not recognize paths; substituting a drive letter for the path allows the application to access files.

d1:

    specifies the drive letter that will refer to the drive or path. d1: cannot be the default drive, and cannot be greater than the value of LASTDRIVE.

d2:path

    specifies the drive or path that will be "renamed" as d1:. The path you specify must start from the root directory.

    d1: and d2: must be different; neither d1: nor d2: can be network drives.

/D

    disables the substitution. You must specify the substitution drive letter, d1:.

Type the SUBST command without parameters to display current substitutions.

You cannot SUBST a network drive. If you attempt such a substitution, the message "Cannot SUBST a network drive" is displayed.

While a substitution is in effect, use the commands CHDIR, MKDIR, RMDIR, and PATH with caution. **Do not** use the following commands while a substitution is in effect: ASSIGN, BACKUP, FDISK, DISKCOMP, DISKCOPY, FORMAT, LABEL, RESTORE, and JOIN.

System Copy (SYS)

SYS d:⏎

SYS is an external command that transfers the MS-DOS system files from the disk in the default drive to the disk in the specified drive.

d:
is the drive letter of the drive to which you want to transfer the MS-DOS system files.

SYS is usually used to update the system or to place the system on a formatted disk that does not contain any files. You must specify a destination drive.

If IO.SYS and MSDOS.SYS are on the destination disk, they must take up the same amount of space on the disk as the new system will need. This means that you cannot transfer system files from an MS-DOS 3.1 disk to an MS-DOS 1.1 disk. You must reformat the MS-DOS 1.1 disk with the FORMAT command before the SYS command can work.

The destination disk must be blank or already have the system files IO.SYS and MSDOS.SYS.

The IO.SYS file is copied first, followed by MSDOS.SYS. IO.SYS and MSDOS.SYS are hidden files that do not appear when you use the DIR command. COMMAND.COM (the command processor) is not transferred by SYS. You must use the COPY command to transfer COMMAND.COM.

Print End of File (TAIL)

TAIL [/n] [/B] [files or directories] ↵

TAIL is an external command that displays the last **n** lines of a file or list of files.

/n

is the number of lines at the end of the file(s) you want to display. If you do not include a number, TAIL prints the last 10 lines. The maximum value is approximately 450. If a line exceeds 130 characters, TAIL truncates it.

/B

indicates that **n** is a value in bytes rather than lines. The default for **n** is 512 if /B is used.

files or directories

is a list of subdirectory names and/or filenames to be operated on. If you do not include files or directories, TAIL gets its input from the keyboard or from redirected input.

For example, this command lists the last 3 lines of every file with .LST extension in the current directory:

tail /3 *.lst ↵

Either of the next two commands lists the last 10 lines of the file ACCOUNTS. The < symbol in the first command indicates that ACCOUNTS is being redirected as input to the TAIL command.

tail < accounts ↵

tail accounts ↵

TIME

TIME [hh][:mm][:ss] ↵

TIME is an internal command that sets the time for the operating system. If you enter the command without parameters, the current time is displayed and you are prompted for a new time. Press the Enter key if you do not want to change the time. For example:

```
A>time↵
Current time is 13:01:10.00
Enter new time:9:05↵
```

The new time known to the operating system is now 09:05:00.00. The colon separates the hours, minutes, and seconds. The period separates the hundredths of a second.

hh
    represents the hour. Use 00 to 23.

mm
    represents the minutes. Use 00 to 59.

ss
    represents the seconds. Use 00 to 59. You do not have to type the seconds or hundredths of a second.

If you enter the TIME command with valid parameters (numbers, not letters, in the indicated range for each parameter), MS-DOS displays the system prompt.

If the parameters are invalid, such as letters instead of numbers or a larger than acceptable range of numbers, MS-DOS displays

```
Invalid time
Enter new time:_
```

MS-DOS also displays this message if you use invalid delimiters, such as a hyphen instead of a colon or a period.

List Directory Paths (TREE)

TREE [d:] [/F]↵

TREE is an external command that lists all directory paths found on the specified drive. TREE can also list all the files in each directory. If you do not specify a drive letter with the TREE command, the default drive is used.

d:

is the name of the drive for which you want to list all the directory paths.

/F

lists the names of the files in each subdirectory.

If you enter the TREE command without any parameters, a listing similar to the following is displayed:

```
C>tree↵

DIRECTORY PATH LISTING FOR VOLUME BUSINESS

Path: \MEMOS

Sub-directories:    AUGUST
                    OCTOBER


Path: \MEMOS\AUGUST

Sub-directories:    None


Path: \MEMOS\OCTOBER

Sub-directories:    None
```

6

If you enter the TREE command with the /F option, the preceding listing is expanded to include all the files in each subdirectory:

```
C>tree /f↵

DIRECTORY PATH LISTING FOR VOLUME BUSINESS

Path: \MEMOS

Sub-directories:    AUGUST
                    OCTOBER

Files:              None


Path: \MEMOS\AUGUST

Sub-directories:    None

Files:              JACKSON  .LET
                    SMITH    .MEM


Path: \MEMOS\OCTOBER

Sub-directories:    None

Files:              CARLSON  .MEM
                    THOMAS   .LET
                    BOYD     .MEM
```

Type File (TYPE)

TYPE [path]filename.ext ⏎

TYPE is an internal command that displays the contents of text files.
You cannot use TYPE with .COM or .EXE files because they contain
non-alphanumeric characters such as Control or Escape sequences.
Tabs expand to 8 characters (column 8, 16, and so on through column
80).

path
is the path through the subdirectory system to the file you want to
display.

filename.ext
is the name of the file to be displayed.

For example, to display the contents of the JOBS.LST file in the
MGR1.NAM\JOBS subdirectory, enter

type \mgr1.nam\jobs\jobs.lst ⏎

Report Repeated Lines (UNIQ)

UNIQ [/U] [/D] [/C]↵

UNIQ is a filter command that reads the standard input and compares adjacent lines. Note that UNIQ compares only neighboring lines, so the file must be sorted (using the SORT command) for UNIQ to detect duplications. You determine the output by setting one or more of the following switches:

/U

lists only unique lines (lines that are not duplicated).

/D

lists only one copy of the duplicated lines.

/C

lists each line with a count of how many times it occurs.

The default for UNIQ is /U and /D; if you specify either /U or /D in a command, the other is automatically turned off. /C supersedes /U and /D. If you specify /C, output is normal, and a count appears before each line indicating the number of times that the line occurs.

In the following example, FGREP finds all files ending in ".p" and having the word "min" or "max" in them, and puts their contents in the JUNK file. Next the SORT command sorts JUNK. UNIQ then lists on the screen each occurrence of a line containing "max" or "min", but lists duplicated lines only once.

```
fgrep /n min *.p > junk↵
fgrep /n max *.p >> junk↵
sort < junk | uniq↵
```

For more information, refer to the FGREP and SORT commands in this chapter.

Version of (VER)

VER⏎

VER is an internal command that displays the version of MS-DOS that you are using. For example, if you enter

ver⏎

MS-DOS displays the following, where **x.y** is the version number of the operating system:

```
MS-DOS Version   x.y
```

Verify Write to Disk (VERIFY)

VERIFY [ON]↵

   or

VERIFY [OFF]↵

VERIFY is an internal command that verifies whether subsequent data written by MS-DOS onto the disk can be read by MS-DOS when requested. VERIFY has the same function as the V switch on the COPY command. VERIFY without parameters displays the current setting (ON or OFF) of the command. For example, if you type

verify↵

MS-DOS displays

```
VERIFY is xx
```

where xx is "on" or "off."

ON

   tells MS-DOS to verify the data each time it writes data to the disk. MS-DOS issues an error message if it cannot do the write operation. When VERIFY is ON, any command involving a write to the disk performs slower because of the extra time involved to verify the operation.

OFF

   turns the verify instruction off.

Display Volume ID (VOL)

VOL [d:] [/C]⏎

VOL is an internal command that displays the Volume ID of a diskette or fixed disk volume. A Volume ID is a name or label you give to a diskette during formatting, or to a fixed disk volume during initialization. It can be up to 11 characters long and uses the valid characters for filenames. MS-DOS displays the Volume ID for a diskette or volume when you enter a DIR, CHKDSK, or VOL command.

d:

is the name of the drive whose ID you want to check. If you do not specify a drive, MS-DOS displays the ID for the default drive.

/C

is a switch you can use to add or change the Volume ID.

If you try to display the Volume ID for a drive that does not have an ID, MS-DOS displays

```
Volume ID for drive X: has not been set.
```

where **X** is the name of the drive.

If you type

vol b: /c⏎

MS-DOS displays the Volume ID for the diskette in drive B, and prompts you for a new Volume ID:

```
Volume ID for drive B: is ARCHIVE1
Enter new Volume ID:_
```

After you enter a new Volume ID and press Enter, MS-DOS writes the ID to the diskette. You can display the new Volume ID for the diskette in drive B by typing

vol b:⏎

Word and Line Count (WC)

WC [/R] [/W] ↵

  or

WC [/R] [/W] [path...] ↵

  or

WC [/R] [/W] [filename.ext...] ↵

WC is an external command that counts words and/or lines in a file or files. WC can count all the files in a directory or in a hierarchy of directories. WC lists the number of lines, the number of words, the filename, and the total count of lines and words. Entering WC with no parameters counts the lines and words in the keyboard input (or redirected input) that follows, up to end-of-file (Ctrl-Z Enter).

/R

is a command switch that expands the count to include files in all subdirectories below the directory named in the WC command. To use this switch, you must list a directory, not a file, in your WC command.

The following example uses the /R switch with the root directory (\) to count the words and lines in all the files in every directory on the default drive:

```
A>wc /r \↵
        21 L        386 W   \MSDOS.SYS
        43 L        495 W   \COMMAND.COM
         5 L         16 W   \CONFIG.SYS
        86 L        455 W   \SUBDIR2\CHAP2.DOC
         . .          .  .   .   .   .
         . .          .  .   .   .   .
      7012 L      51311 W   total
```

/W

is a command switch that omits the word count from the WC operation. Counting only lines is faster than counting both words and lines.

path

is the path through the subdirectory system to the file or subdirectory you want to count. If you list a directory name (or \ to indicate the root directory) without a filename, WC counts all the files in that directory.

filename.ext

is the name of the file you want to count. You can use wildcard characters to refer to more than one file.

The first example below counts lines and words in the FINAL.MEM file, and the second example counts lines in the FINAL.MEM and FINAL.BAK files.

wc final.mem ⏎

wc /w final.mem final.bak ⏎

You can list multiple files and directories in a WC command, and the files and directories can be on different drives. The following command counts the number of lines in all files with .DAT extension on the default drive and all files in the subdirectory DRAFT1 on drive B:

wc /w *.dat b:\draft1 ⏎

MS-DOS Messages

This chapter lists many of the messages you might see while you work with your computer. The messages are shown in boldface and are listed alphabetically. An explanation follows each message. Sometimes you are given a reason for why you are seeing that message and what you can do to successfully complete the procedure you want. The beginning of the explanation lists which program(s) you might be using when you receive the message.

When MS-DOS encounters a problem, it displays an error message on the screen. Different error messages are displayed for different types of errors. Error messages are displayed in situations like these:

▶ You entered a command that MS-DOS does not recognize.

▶ You misspelled a filename, or you entered an incorrect drive name.

▶ You inserted a diskette that does not contain the file or program MS-DOS is looking for.

If you see the word "filename" or "xxxx" in brackets or parentheses in one of the messages, it means that the message you see on your screen contains the specific filename or number(s) that are pertinent to the operation you are attempting.

Abort edit (Y/N)?

EDLIN: This message is displayed when you choose the Q (Quit) command in EDLIN. The Quit command exits the editing session without saving any editing changes. Specify Y for yes or N for no.

Abort, Retry, Ignore?

MS-DOS: If a disk or device error occurs at any time during a command or program, MS-DOS returns this message and asks you to abort the command or program, retry it, or ignore the error.

Access denied

COMMAND: The command processor cannot invoke the requested command without violating the access mode of the file, subdirectory, or device involved.

All files cancelled by operator

PRINT: This message is displayed when you specify the /T switch with the PRINT command.

All specified files are contiguous

CHKDSK: All files are allocated contiguously on the disk without fragmentation.

Allocation error in file, size adjusted

CHKDSK: An invalid sector number was found in the FAT. The file was truncated at the end of the last valid sector.

Amount read less than size in header

EXE2BIN: The program portion of the file is smaller than the file header indicates.

Are you sure (Y/N)?

COMMAND: MS-DOS displays this message if you try to delete *.* (all files in the current directory). Specify Y for yes or N for no.

Attempted write protect violation

FORMAT: The diskette you are formatting is write-protected and cannot be written on.

Backing up files to target drive
Target Number:

BACKUP: This message displays the sequence of files being backed up.

Backup file sequence error

RESTORE: A file to be restored is backed up on more than one diskette and you have not inserted the diskette that contains the beginning of the file. Insert the correct diskette and run RESTORE again.

Bad call format reading drive (x:)

Device error: See Appendix B.

Bad call format writing drive (x:)

Device error: See Appendix B.

Bad command error reading drive (x:)

Device error: See Appendix B.

Bad command error writing drive (x:)

Device error: See Appendix B.

Bad command or file name

COMMAND: The command processor cannot find the file you asked it to run. Either you mistyped the filename or the file does not exist on the disk. Check to see that you are in the right directory and that you have specified the path correctly.

Bad drive specifier on command line

DISKCOPY: You entered a drive specifier that is not a letter.

Bad or missing (filename)

MS-DOS: You specified an invalid device in the CONFIG.SYS file. Check the accuracy of the device statement in CONFIG.SYS.

Bad or missing Command Interpreter

MS-DOS: MS-DOS either could not find or could not read the COMMAND.COM file; either the file is missing from the root directory, or the file is invalid. If rebooting the system fails, boot using a backup system diskette and copy the COMMAND.COM file from your backup MS-DOS system diskette onto the diskette used to start MS-DOS.

You can also receive this message if COMMAND.COM has been moved from the directory it was originally in when you started MS-DOS.

Bad unit error reading drive (x:)

Device error: See Appendix B.

Bad unit error writing drive (x:)

Device error: See Appendix B.

BREAK is off (or on)

BREAK, COMMAND: This message tells you the current setting of BREAK.

Cannot CHDIR to (filename) - tree past this point not processed

CHKDSK: CHKDSK is traveling the tree structure of the directory and is unable to proceed to the specified directory. Subdirectories beneath this directory are not verified.

Cannot CHDIR to root
Processing cannot continue

CHKDSK: CHKDSK is traveling the tree structure of the directory and is unable to return to the root directory. CHKDSK cannot continue checking the remaining subdirectories to the root.

Cannot copy high-density diskettes to a standard floppy drive

DISKCOPY: You can copy a high-density diskette only to a high-capacity drive.

Cannot discover diskette type

DISKCOPY: The source diskette is not formatted correctly and the system cannot determine the source diskette type.

Cannot DISKCOMP to or from a Network drive

DISKCOMP: You cannot use DISKCOMP to compare files on a network drive or a drive that is part of your computer but is shared on the network.

Cannot DISKCOPY to or from a Network drive

You cannot use DISKCOPY to copy files to or from a network drive or a drive that is on your computer but is shared by the network. Use COPY *.* instead of DISKCOPY.

Cannot do binary reads from a device

COMMAND: This message appears during COPY command processing. The COPY cannot be done in binary mode when you are copying from a device. Remove the /B switch or specify an ASCII copy with the /A switch.

Cannot edit .BAK file--rename file

EDLIN: You attempted to edit a backup copy created by EDLIN. Either rename the file or copy the .BAK file and give it a different extension.

Cannot FORMAT a Network drive

FORMAT: You cannot use the FORMAT command to format a network drive or a drive on your computer that is shared on the network.

Cannot format a SUBSTed or ASSIGNed drive

FORMAT: You cannot use the FORMAT command to format a drive that is part of a SUBST or an ASSIGN.

Cannot format track

DISKCOPY: DISKCOPY could not format a track on the target diskette. Run DISKCOPY again or try another diskette.

Cannot JOIN to a Network drive

JOIN: You cannot use the JOIN command to join a local drive to a network drive or to join a network drive to a local drive.

Cannot LABEL a Network drive

LABEL: You cannot create a new volume label or change an existing volume label on a redirected block device.

Cannot open (filename)

PRINT: Either MS-DOS cannot find the specified file to print or the file does not exist. Check the command for a valid filename.

Cannot recover . entry, processing continued

CHKDSK: The . entry (current directory) is defective.

Cannot recover .. entry

CHKDSK: The .. entry (parent directory) is defective.

Cannot RECOVER to a Network drive

RECOVER: You cannot use the RECOVER command to recover files from a network drive or a drive on your computer that is shared on the network.

Cannot SUBST to a Network drive

SUBST: You cannot use the SUBST command to substitute a drive for a network path or substitute a network path for a local path.

Cannot SYS to a Network drive

SYS: You cannot use the SYS command to transfer system files to a network drive or to a drive on your computer that is shared on the network. Use PAUSE to pause the server, invoke the SYS command, and then use CONTINUE to continue the server.

CHDIR .. failed, trying alternate method

CHKDSK: In traveling the tree structure, CHKDSK could not return to a parent directory. It tries to return to that directory by starting over at the root and traveling down.

COMn: xxxx,p,d,s,t

MODE: This message indicates the MODE command setting of either COM1 or COM2. The individual parameter values are as follows:

n can be either 1 or 2 and indicates whether the setting is for COM1 or COM2.

xxxx is the baud rate.

p is the parity (even, odd, or none).

d is the number of data bits per character. The number is 8 with no parity or 7 with even or odd parity.

s is the number of stop bits (1 or 2).

t is the type of serial device (p for serial printer or - for any other serial device).

Compare error at offset

COMP: This message indicates that the files being compared contain different values at the displayed offset into the file.

Compare process ended

DISKCOMP: This message indicates that the DISKCOMP utility has completed the comparison.

COM: ,e,7,1
Compare error(s) on
Track ,side

DISKCOMP: One or more locations on the indicated track and side contain different information.

Compare more diskettes (Y/N)?

DISKCOMP: This message indicates that DISKCOMP has finished comparing the specified diskettes and that by pressing Y you can compare additional diskettes without invoking DISKCOMP again.

Compare more files (Y/N)?

COMP: This message indicates that COMP has finished comparing the specified files and that by pressing Y you can compare additional files without invoking COMP again.

Content of destination lost before copy

COPY: A file to be used as a source file in the COPY command has been overwritten before the copy is completed. For example, if you enter this command, MS-DOS destroys FILE2 before it can be copied:

    copy file1 + file2 file2↵

Convert lost chains to files (Y/N)?

CHKDSK: If you respond Y to this prompt, CHKDSK recovers the lost blocks it found when checking the diskette. CHKDSK creates a directory entry and a file for you with the filename FILEnnnn. If you respond N, CHKDSK frees the lost blocks so they can be re-allocated.

Copy another (Y/N)?

DISKCOPY: Respond Y if you want to copy another diskette; respond N if you do not want to copy another diskette.

Create another diskette using data from the fixed disk (Y/N)?

DISKCOPY: Answer Y (yes) if you want DISKCOPY to create another diskette using the source information it read previously. Answer N (no) to return to MS-DOS.

Data error

DISKCOPY: A data error was encountered in reading the source diskette. Retry the DISKCOPY command, making sure you have specified the correct parameters. If DISKCOPY still cannot read the diskette, either the diskette or the drive might be bad.

Data error reading drive (x:)

Device error: See Appendix B.

Data error writing drive (x:)

Device error: See Appendix B.

Destination diskette is write-protected

DISKCOPY: DISKCOPY cannot write to the destination diskette because it is write-protected. Remove the write-protect tab from the diskette and run DISKCOPY again.

Directory is totally empty, no . or ..

CHKDSK: The specified directory does not contain references to current and parent directories. Delete the specified directory and recreate it.

Disk error reading drive (x:)

Device error: See Appendix B.

Disk error reading FAT (x)

CHKDSK: One of your File Allocation Tables has a defective sector in it. MS-DOS automatically uses the other FAT, but it is a good idea to copy all your files to another diskette.

DISK error while reading drive A
Abort, Ignore, Retry?

DISKCOPY: Disk errors were encountered during DISKCOPY processing. Refer to Appendix B for more information.

Disk error writing drive (x:)

Device error: See Appendix B.

Disk error writing FAT (x:)

CHKDSK: One of your File Allocation Tables has a defective sector in it. MS-DOS automatically uses the other FAT. It is a good idea to copy all your files onto another diskette.

Diskette/Drive not compatible

DISKCOPY, DISKCOMP: The destination diskette/drive is different from the source diskette/drive and prohibits the copy or comparison.

Disk full--write not completed

EDLIN: You gave the END command, but the diskette did not contain enough free space for the file. EDLIN aborted the E command and returned you to the operating system. Part of the file might have been written to disk and saved. Delete the saved portion and restart the editing session. The file will not be available after this error.

Disk unsuitable for system drive

FORMAT: FORMAT detected a bad track on the disk where system files should reside. You should store only data files on the disk you tried to format when this message was displayed.

Divide overflow

MS-DOS: The microprocessor has set the divide overflow flag which is usually caused by division by zero. This message usually indicates that an unrecoverable program error has occurred.

(.)(..) Does not exist

CHKDSK: Either the . or .. directory entry is invalid.

Do you see the leftmost 0? (Y/N)

MODE: You specified MODE R,T to align the display. MODE continues to display the message until you type Y. Each time you type N, MODE shifts the display.

Do you see the rightmost 9? (Y/N)

MODE: You specified MODE L,T to align the display. MODE continues to display the message until you type Y. Each time you type N, MODE shifts the display.

Do you want to try again? (Y/N)

DISKCOPY: An error occurred while DISKCOPY was reading or writing a diskette. This option lets you retry the operation that caused the error. If you answer N (no), you exit the DISKCOPY utility.

Drive not ready error

DISKCOPY: DISKCOPY could not continue because the specified drive is not ready. Check to see whether the diskette is in the drive and the drive door is closed.

Duplicate file name or File not found

COMMAND, 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)

COMMAND: This message tells you the current status of ECHO.

End of input file

EDLIN: The entire file was read into memory. If the file is read in sections, this message indicates that the last section of the file is in memory.

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 indicating which subdirectory is invalid. If you specified the /F switch, CHKDSK attempts to correct the error.

Error in .EXE file

COMMAND: The .EXE file you asked MS-DOS to load has an invalid internal format.

Error opening overflow file on fixed disk

DISKCOPY: DISKCOPY could not open an overflow file in the fixed disk volume you specified. Perhaps the directory is full, other files have been left open, or the fixed disk is defective. Check the fixed disk and try again.

(type of error) error reading file

Device error: See Appendix B.

Error reading overflow file

DISKCOPY: A data error occurred while DISKCOPY was reading the overflow file on the fixed disk. Run DISKCOPY again. If the error persists, run CHKDSK on the fixed disk.

Error reading source File Allocation Table

DISKCOPY: The File Allocation Table on the source diskette has information DISKCOPY needs to do the copying and DISKCOPY cannot read it. Try DISKCOPY again. If the error persists, your source diskette is unusable. Use the command COPY *.* to save as many files as possible.

Error trying to open backup log file. Continuing without making log entries.

BACKUP: BACKUP is unable to open the backup log file. There may be too many open files.

Error writing File Allocation Table to target diskette

DISKCOPY: After formatting the target diskette, DISKCOPY tried to put a new File Allocation Table on it and failed. The File Allocation Table allows MS-DOS to keep track of file space usage. Without it, the diskette is unusable. Try running DISKCOPY again. If the DISKCOPY fails, use another target diskette.

Error writing label to target diskette

DISKCOPY: DISKCOPY was writing identifying information to the new target diskette when an error occurred. This error condition makes the target diskette unusable. Run DISKCOPY again; if the error persists, discard the diskette and use a new one.

Error writing overflow file

DISKCOPY: An error occurred while DISKCOPY was writing the overflow file. Perhaps the fixed disk is full. Check the fixed disk and run DISKCOPY again.

Error writing to device

COMMAND: You tried to send too much data to a device. MS-DOS could not write the data to the specified device.

Errors found, F parameter not specified

CHKDSK: Corrections will not be written to disk. CHKDSK found errors on the diskette. If you did not specify the /F switch, CHKDSK continues printing messages, but does not correct the errors.

Errors on list device indicate that it might be off-line. Please check it.

PRINT: Your printer is off-line.

EXEC failure

COMMAND: Either MS-DOS found an error when reading a command or the FILES statement in the CONFIG.SYS file is set too low. Increase the value and restart MS-DOS.

File allocation table bad

COMMAND: The diskette might be defective. Run CHKDSK to check the diskette.

File allocation table bad drive (x:)

CHKDSK: The diskette might be defective. Run CHKDSK to check the diskette.

File cannot be converted

EXE2BIN: The input file is not in the correct format. CS:IP does not meet the criteria specified, or it meets the .COM file criterion but has segment fixups. This message is also displayed if the file is not a valid executable file.

File cannot be copied onto itself

COMMAND: The source filename you specified is the same as the destination filename. For example, you entered

   copy file1 file1 ↵

The file copy must be renamed if it is on the same drive, or it must be on a different drive than the source file.

File creation error

COMMAND: You tried to add a new filename or replace a file that already exists in the directory. If the file already exists, it is a read-only file and cannot be replaced. Run CHKDSK on the diskette to determine the cause of the error.

EXE2BIN: EXE2BIN cannot create the output file. Run CHKDSK to determine if the directory is full, or if some other condition caused the error.

(filename) contains non-contiguous blocks

CHKDSK: The filename specified is not allocated contiguously on the diskette. If you specify the /F switch, CHKDSK fixes this error.

(filename) file not found

PRINT: You switched diskettes while a file was queued up, but before it started to print. Reissue the PRINT command for that filename.

(filename) is cross-linked on cluster

CHKDSK: Make a copy of the file you want to keep, and then delete both files that are cross-linked.

Filename must be specified

EDLIN: You did not specify a filename when you started EDLIN.

File not found

COMMAND, EXE2BIN, FIND, EDLIN, RECOVER: MS-DOS cannot find the file that you specified. Check that the pathname is accurate and that the file exists in the directory you specified.

Files cannot be added to this diskette unless the PACK (/P) switch is used
Set the switch? (Y/N)

BACKUP: You are adding files to a packed diskette and can add only files that are packed. If you answer Y (yes), BACKUP sets the /P switch and continues. If you answer N (no), you exit the BACKUP utility.

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.

Fixed disk drive letter expected

DISKCOPY: You entered the /O switch without adding a fixed disk drive specifier.

Fixups needed - base segment (hex:)

EXE2BIN: The source (.EXE) file contained information indicating that a load segment is required for the file. Specify the absolute segment address at which the finished module is to be located.

FOR cannot be nested

COMMAND: Nesting FOR statements is not allowed in batch files.

Format another (Y/N)?

FORMAT: Type Y (for yes) to format another diskette. Type N (for no) if you do not want to format another diskette. If you accidentally type Y, you can abort the format process by typing Ctrl-Break in response to the "Strike any key to begin formatting" message.

Format failure

FORMAT: MS-DOS could not format the diskette. This message is always displayed with the reason MS-DOS could not format the diskette.

Formatting target diskette...

DISKCOPY: The target diskette is blank, or you specified the /F switch. DISKCOPY is now formatting the entire diskette before writing data to it.

General drive failure error

DISKCOPY: Either the source drive or the target drive has caused a read or write error. One of the diskettes may be unreadable.

Illegal device name

MODE: You have specified an invalid device name. The device name must be LPT1, LPT2, LPT3, COM1, or COM2.

Incompatible system size

SYS: The system files IO.SYS and MSDOS.SYS occupy more space on the source diskette than is available on the destination diskette.

Incorrect MS-DOS version

ASSIGN, BACKUP, CHKDSK, EDLIN, FIND, FORMAT, MORE, PRINT, RECOVER, RESTORE, SORT, SYS: Many utilities cannot run on earlier versions of MS-DOS. Some utilities can run only on the version of MS-DOS for which they were configured.

Incorrect parameter

SHARE: A parameter specified in the SHARE command is invalid.

Infinite retry on parallel printer timeout

MODE: You have specified the P parameter which requests continuous retry on timeout errors.

Insert diskette with batch file and press any key when ready

COMMAND: The diskette containing the batch file you specified is no longer in the drive you originally specified. Re-insert the diskette that contains the batch file in the appropriate drive, or press Ctrl-C.

Insert MS-DOS diskette in drive (x:) and strike any key when ready

FORMAT: You specified FORMAT /S, 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 (a bootable diskette) in the drive specified.

Insert new diskette for drive (x:) and strike any key when ready

FORMAT: Insert a blank diskette into the appropriate drive and press any alphanumeric key to begin formatting. **CAUTION:** If there is any data on the diskette, it will be destroyed by the format process.

Insert source diskette in drive (x:)

DISKCOPY: Insert 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 bootable diskette from which to read the IO.SYS and MSDOS.SYS files. Insert a bootable diskette into the specified drive and press any alphanumeric key to start the system copy.

Insert target diskette in drive (x:)

DISKCOPY: Your source and destination drives are the same. Re-insert the destination diskette into the specified drive.

Insufficient disk space

COMMAND, EXE2BIN, SORT: The diskette is full. It does not contain enough room to perform the specified operation.

Insufficient memory

BACKUP, CHKDSK, EDLIN, EXE2BIN, RESTORE, SORT, SYS: There is not enough memory to perform the specified operation.

Insufficient memory for copying

DISKCOPY: DISKCOPY sought more memory so that it could put information on the diskette. MS-DOS returned a memory value insufficient to allow DISKCOPY to proceed. Install more memory, reduce the number of buffers and/or device drivers, and take out any memory resident programs.

Insufficient memory for system transfer

FORMAT: Your memory configuration is insufficient to transfer the MS-DOS system files IO.SYS and MSDOS.SYS.

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 to make room for the lost files.

Intermediate file error during pipe

COMMAND: The pipe operation makes use of temporary files on the diskette which are automatically deleted after the piping process is complete. An error has occurred in one of these files.

Invalid baud rate specified

MODE: You specified an invalid baud rate.

Invalid characters in volume label

FORMAT: The volume label can contain only up to eleven alphanumeric characters.

Invalid COMMAND.COM
Insert COMMAND.COM disk in default drive and strike any key when ready

COMMAND: MS-DOS needs to reload COMMAND.COM from disk; however, MS-DOS cannot find COMMAND.COM on the diskette or the copy found is invalid. Insert a diskette into the default drive that contains a copy of COMMAND.COM identical to the version on the diskette with which you started MS-DOS.

Invalid COMMAND.COM
Insert COMMAND.COM disk in drive (x:) and strike any key when
ready

COMMAND: MS-DOS needs to reload COMMAND.COM from disk; however, MS-DOS cannot find COMMAND.COM on the diskette or the copy found is invalid. Insert a diskette into the specified drive that contains a copy of COMMAND.COM identical to the version on the diskette with which you started MS-DOS.

Invalid country code

MS-DOS: You specified a country number in your CONFIG.SYS file that is not in the table of files configured in this implementation of MS-DOS. Country codes must be in the range 1–99. See the COUNTRY command in Section 6.4 for more information on the supported countries.

Invalid current directory

CHKDSK: Your diskette is bad. Replace the diskette or make another copy from your backup system diskette.

Invalid date

COMMAND, DATE: You specified an invalid date in response to the date prompt when starting MS-DOS.

Invalid directory

COMMAND: The directory you specified either does not exist or is invalid. Check to see that you entered the directory name correctly.

Invalid drive in search path

COMMAND: The drive does not exist.

Invalid drive or file name

EDLIN, RECOVER: Specify a valid drive or a valid filename.

Invalid drive specification

BACKUP, CHKDSK, COMMAND, DISKCOPY, FORMAT, SYS, RESTORE: You specified an incorrect drive letter. Specify a valid drive.

Invalid media or Track 0 bad - disk unusable

FORMAT: FORMAT could not format the diskette. If successive attempts fail, discard the diskette.

Invalid number of parameters

BACKUP, COMMAND, FIND, RECOVER, RESTORE: You have specified the wrong number of options in the command line.

Invalid parameter

BACKUP, CHKDSK, COMMAND, EDLIN, FIND, FORMAT, MODE, PRINT, RESTORE: One or more of the switches or parameters you specified is wrong.

Invalid path

BACKUP, RESTORE: You specified an invalid path. The path does not specify a unique directory.

Invalid path or file name

COPY: Specify a valid pathname or filename.

Invalid path, not directory, or directory not empty

COMMAND: You cannot remove the directory requested for one of the specified reasons.

Invalid subdirectory entry

CHKDSK: The subdirectory you specified either does not exist or is invalid. Check that you entered the subdirectory name correctly.

Invalid switch character

VDISK: VDISK found a slash (/) that was not followed by an E in the DEVICE = VDISK.SYS part of CONFIG.SYS. VDISK tries to install the virtual disk in low memory.

Invalid time

COMMAND, TIME: You specified an invalid time in response to the time prompt when starting MS-DOS.

Label not found

COMMAND: There is a GOTO command to a nonexistent label in a batch file.

Last file not backed up

BACKUP: The last file specified was not backed up because the diskette became full. The file was deleted because not all of it could be backed up.

Line too long

EDLIN: During a Replace command, the string you gave as the replacement caused the line to expand beyond 253 characters. Divide the long line into two lines and retry the Replace command.

List output is not assigned to a device

PRINT: When you first run PRINT, it asks you what device you want to specify as a print spooler. This message appears if PRINT is set up for a nonexistent device.

LPT#: not redirected

MODE: Output is now directed to the parallel printer instead of being redirected to a serial device.

LPT#: redirected to COMn:

MODE: Output that would normally go to the parallel printer is now redirected to serial device COM1 or COM2.

LPT#: set for 80

MODE: You attempted to set the parallel printer line length to 80 characters.

LPT#: set for 132

MODE: You attempted to set the parallel printer line length to 132 characters.

Memory allocation error. Cannot load COMMAND, system halted.

COMMAND: Restart MS-DOS. If this error persists, make a new copy of the MS-DOS diskette from your backup copy of the system diskette.

--More--

MORE: Press the Spacebar to view more of the file or directory.

Must specify destination line number

EDLIN: You must specify a destination line number when you are copying and inserting lines with EDLIN.

Must specify ON or OFF

COMMAND: The command requires an argument, either ON or OFF.

Name of list device [PRN]:

PRINT: This prompt appears when you run PRINT for the first time in a computer session. You can specify any valid device which then becomes the PRINT output device.

New file

EDLIN: EDLIN did 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 that you correctly typed the filename of the file you want to edit.

No files match (d:filename)

PRINT: You gave a filespec for files to add to the queue, but no files match the specification.

No free file handles
Cannot start COMMAND.COM exiting

COMMAND: Restart MS-DOS. If this message persists, increase the FILES parameter in the CONFIG.SYS file.

No paper error writing device (dev)

Device error: See Appendix B.

No path

COMMAND, PATH: There is no current command search path.

No retry on parallel printer timeout

MODE: The P parameter has been omitted, so there is no request for retry on timeout errors.

No room for system on destination disk

SYS: There is not enough room for the system files on the destination diskette. Delete some files to make room for the system files or use another diskette. You might want to use a freshly formatted diskette.

No room in directory for file

EDLIN: You tried to save a file to the root directory but it is full. Subdirectories are not limited in size as is the root directory.

No room in root directory

LABEL: An error has occurred while creating the volume label; probably there is no room in the root directory for another entry. Try deleting a file from the root directory and re-invoking the LABEL command.

Non-MS-DOS disk error reading drive (x)

Device error: See Appendix B.

Non-System disk or disk error
Replace and strike any key when ready

FORMAT: You specified /S and either the default drive did not have a system on it, or an error was found in reading the system from it.

Not able to back up file

BACKUP: The specified file cannot be backed up due to a file sharing conflict. Try again and specify the /M switch.

Not able to restore file

RESTORE: The specified file cannot be opened due to a file sharing conflict.

Not enough memory

SHARE: The available memory is less than what SHARE requires; SHARE terminates without being installed.

Not enough room to merge the entire file

EDLIN: There was not enough room in memory to hold the file during a Transfer command. You must free some memory by writing some files to disk or deleting some files before you can transfer this file.

Not found

EDLIN: You specified a Search or a Replace command and EDLIN could not find another occurrence of the specified string.

Not ready error reading drive (x:)

Device error: See Appendix B.

Not ready error writing drive (x:)

Device error: See Appendix B.

O.K.?

EDLIN: This prompt occurs during Search and Replace command processing. If you press any key except Y or Enter, the search or replace continues.

Only two drive specifiers allowed on command line

DISKCOPY: You entered more than two drive specifiers. Retype the command line with only one source drive specifier and one target drive specifier.

Out of environment space

COMMAND: There is not enough room in the program environment to accept more data.

Parameter not compatible with fixed disk

FORMAT: The parameter you specified cannot be used with the fixed disk (/B or /V).

Press any key to begin formatting (x:)

FORMAT: This prompt is issued before you format a diskette. Press any alphanumeric key to begin formatting. To discontinue the format, press Ctrl-Break.

Press any key to begin recovery of the (xxx) file(s) on drive (x:)

RECOVER: This prompt is issued before you recover a diskette or file. Press any alphanumeric key to begin recovering. To discontinue the recovery, press Ctrl-Break.

Press any key when ready

DISKCOPY: This prompt occurs when you are copying diskettes. When you have inserted the diskettes into the appropriate drives, press any alphanumeric key to begin copying the diskette. To discontinue the diskette copy, press Ctrl-Break.

PRINT queue is empty

PRINT: There are no files waiting to be printed.

PRINT queue is full

PRINT: There is room for only 10 files in the list of files waiting to be printed.

Printer error

MODE: The print device generated an error when output was sent to it.

Printer lines per inch set

MODE: You attempted to set the vertical spacing on the printer.

Probable non-MS-DOS disk
Continue (Y/N)?

CHKDSK: The diskette you are using is not recognized by this version of MS-DOS. The diskette either was created by another system with a format that is not supported on this version of MS-DOS or is not an MS-DOS diskette. Do not continue processing if CHKDSK has returned this message for a removable diskette. If this message is returned for a fixed disk, the information describing the diskette characteristics to MS-DOS has been destroyed. In this case, continue CHKDSK processing. You will probably have to set up your fixed disk again.

Processing cannot continue

CHKDSK: There is not enough memory free in your machine to process CHKDSK for this diskette. You must obtain more memory to run CHKDSK.

Program too big to fit in memory

COMMAND: You must acquire more memory to run your application. Some applications you have run might still be using some memory. You can try to restart MS-DOS; however, if you still receive this message, you must acquire more memory.

Read fault error

DISKCOPY: An error was encountered in reading the specified diskette. Run DISKCOPY again, making sure you use the correct parameters. If DISKCOPY still cannot read the diskette, either the diskette or the drive might be bad.

Read fault error reading drive (x:)

Device error: See Appendix B.

Resident part of PRINT installed

PRINT: This is the first message MS-DOS displays when you issue the PRINT command. It means that available memory has been reduced by several thousand bytes to process the PRINT command concurrent with other processes.

Resident portion of MODE loaded

MODE: This message indicates that a portion of the MODE utility has been loaded in resident memory. This is sometimes necessary when MODE is used for tasks other than screen settings.

Sector not found error

DISKCOPY: An error was encountered in reading the specified diskette. Run DISKCOPY again, making sure you use the correct parameters. If DISKCOPY still cannot read the diskette, either the diskette or the drive might be bad.

Sector not found error reading drive (x:)

Device error: See Appendix B.

Sector not found error writing drive (x:)

Device error: See Appendix B.

Sector size adjusted

VDISK: VDISK adjusted the sector size value in the DEVICE = VDISK.SYS part of the CONFIG.SYS file.

Sector size too large in file (filename)

Device error: 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

DISKCOPY: An error was encountered in reading the specified diskette. Run DISKCOPY again, making sure you use the correct parameters. If DISKCOPY still cannot read the diskette, either the diskette or the drive might be bad.

Seek error reading drive (x:)

Device error: See Appendix B.

Seek error writing drive (x:)

Device error: See Appendix B.

SHARE already installed

SHARE: You attempted to load SHARE after it has already been loaded.

Sharing buffer exceeded

The buffer area for file sharing and block locking has been exceeded.

Source and target drives are the same

BACKUP, RESTORE: You specified the same drive specifier for the source drive and the target drive. BACKUP and RESTORE require you to specify different drives for the source and target.

Source and target drives MUST be the same when you use the overflow option

DISKCOPY: When you use the /O switch, the source and target disk drives must be the same.

Source does not contain backup files

RESTORE: You attempted to restore from a diskette that does not contain files that have been backed up with the BACKUP utility.

Specified COMMAND search directory bad

MS-DOS: The SHELL statement in the CONFIG.SYS file is incorrect. The place you told MS-DOS to find COMMAND.COM does not exist, COMMAND.COM is not in that place, or you have specified the wrong name.

Specified drive does not exist, or is non-removable

DISKCOPY, DISKCOMP: The drive specifier is for a fixed disk drive or does not exist on your computer.

Strike a key when ready...

COMMAND: This prompt occurs during command processing and is always accompanied by another message. This message is also displayed if you have inserted a PAUSE statement in a batch file. Usually, you are asked to insert diskettes into appropriate drives before this prompt. Press any alphanumeric key to begin command processing.

Syntax error

COMMAND, FIND: Check that you typed the command correctly.

System transferred

FORMAT, SYS: The system files MSDOS.SYS and IO.SYS have been transferred during FORMAT or SYS command processing.

Target cannot be used for backup

BACKUP: Back up the files to a different device or restart the system and try again.

Target diskette is write-protected

DISKCOPY: The diskette you are copying to is write-protected. Remove the write-protect tab and try again.

Target is full

RESTORE: The target disk you are attempting to restore files to is full. Erase files from the disk or restore the files to a different device.

Terminate batch job (Y/N)?

COMMAND: If you press Ctrl-Break during batch processing, MS-DOS asks whether or not you want to end batch processing. Press Y to end processing; press N to continue the batch job.

The last file was not restored

RESTORE: The RESTORE routine stopped before it completed restoring the last file listed or there was not enough room to restore all files listed.

The source drive (x:) does not exist

DISKCOPY: The source drive you specified does not exist. Enter a valid drive name.

The source drive (x:) is a fixed disk.
DISKCOPY works only with diskettes.

DISKCOPY: The source drive letter you specified is assigned to a fixed disk. DISKCOPY can copy only one diskette to another.

The target drive (x:) does not exist

DISKCOPY: The target drive you specified does not exist. Enter a valid drive name.

The target drive (x:) is a fixed disk.
DISKCOPY works only with diskettes.

DISKCOPY: The target drive letter you specified is assigned to a fixed disk. DISKCOPY can copy only one diskette to another.

This version of DISKCOPY requires MS-DOS version 2.0 or higher.

DISKCOPY: You cannot run this version of DISKCOPY under MS-DOS versions 1.0 and 1.25.

Track 0 bad - disk unusable

FORMAT: FORMAT can accommodate for defective sectors on the diskette except those near the beginning. FORMAT cannot write the boot record. Use another diskette.

Unable to create a directory

COMMAND: MS-DOS could not create the directory you specified. You might have a file by the same name or the diskette might be full.

Unknown media type error

DISKCOPY: The source diskette is not formatted correctly and the system cannot determine the source diskette type.

Unrecognized command in CONFIG.SYS

MS-DOS: There is an invalid statement in your CONFIG.SYS file.

Unrecognized source diskette type

DISKCOPY: The source diskette is not formatted correctly and the system cannot determine the source diskette type.

Unrecoverable error in directory
Convert directory to file (Y/N)?

CHKDSK: If you respond Y to this prompt, CHKDSK converts the bad directory into a file. You can then fix the directory yourself or delete it.

Unrecoverable file sharing error

SHARE: A file sharing conflict occurred; files cannot be restored.

VDISK not installed - insufficient memory

VDISK: This message occurs for one of three reasons:

▶ There would be less than 64K of memory even after an attempt to adjust the virtual disk size and number of directories.

▶ Your computer does not have expanded memory and you have specified the /E switch.

▶ The amount of available expanded memory is too small to install the virtual disk.

VERIFY is off (or on)

COMMAND, VERIFY: This message tells you the current setting of VERIFY.

Volume in drive (x:) has no label

COMMAND, DIR: This message is displayed in response to the DIR command when you have not specified a volume label.

Volume in drive (x:) is (filename)

COMMAND: This message is displayed in response to the DIR command when you have specified a volume label.

Warning - directory full

RECOVER: The root directory is too full for RECOVER processing. Delete some files in the root directory to free space.

Warning! Diskette is out of sequence
Replace diskette and continue if okay
Strike any key when ready

RESTORE: The diskette you have inserted to be restored is not the correct diskette for the backup sequence. Insert the correct diskette and press any key to continue restoring.

Warning! File (x) is a hidden file
Replace the file (Y/N) ?

RESTORE: Answer Y (yes) if you want the backed up version to replace the previous version.

Warning! File (x) is a read-only file
Replace the file (Y/N) ?

RESTORE: RESTORE cannot restore a read-only file. Answer Y (yes) if you want the backed up version to replace the previous version.

Warning! Files in the target drive (x:) root directory will be erased

BACKUP: This message warns you that the files in the root directory of the target drive will be erased as files are backed up into that directory.

Warning! No files were found to back up

BACKUP: The files on the source diskette do not match the files you specified to be backed up.

Warning! No files were found to restore

RESTORE: The files on the backup diskette do not match the files you specified to be restored.

Warning: Read error in EXE file.

EXE2BIN: The amount read was less than the size of the header. This is a warning message only. EXE2BIN will attempt to continue processing.

Write fault error

DISKCOPY: DISKCOPY is unable to write to the destination diskette. Run DISKCOPY again, making sure you use the correct parameters. If DISKCOPY is still unable to write to the diskette, use another diskette.

Write fault error writing drive (x:)

Device error: See Appendix B.

Write-protect error writing drive (x:)

Device error: See Appendix B.

Write-protect violation

FORMAT: The diskette is write-protected.

/x is not a valid switch

DISKCOPY: The DISKCOPY switch you specified is not one of the valid options.

(x:) - the overflow volume - does not exist or is not responding.

DISKCOPY: The drive you specified for the overflow file either does not exist or is not responding.

(x:) - the overflow volume - is a floppy diskette. It must be a fixed disk.

DISKCOPY: You cannot open an overflow file on a diskette. Retype the command line and specify a fixed disk drive.

(xxxx) of (xxxx) bytes recovered

RECOVER: This message tells you how many bytes MS-DOS was able to recover of the diskette or file.

(filename) cancelled by operator

PRINT: This message is printed on the printer when you specify the /T switch in the PRINT command.

The following messages are displayed when certain MS-DOS commands have completed processing. These are informational messages only.

(xxxx) bytes available on disk

CHKDSK, FORMAT

(xxxx) bytes disk space freed

CHKDSK

(xxxx) bytes disk space would be freed

CHKDSK

(xxxx) bytes free
   COMMAND

(xxxx) bytes in (yyyy) directories
   CHKDSK

(xxxx) bytes in (yyyy) hidden files
   CHKDSK

(xxxx) bytes in (yyyy) recovered files
   CHKDSK

(xxxx) bytes in (yyyy) user files
   CHKDSK

(xxxx) bytes in bad sectors
   FORMAT

(xxxx) bytes total disk space
   FORMAT

(xxxx) bytes total memory
   CHKDSK

(xxxx) bytes used by system
   FORMAT

(xxxx) bytes would be in (yyyy) recovered files
   CHKDSK

(xxxx) file(s)
   COMMAND

(xxxx) file(s) copied

COMMAND

(xxxx) file(s) recovered

RECOVER

(xxxx) lost clusters found in (xxxx) chains

CHKDSK

(filename) contains (xxxx) non-contiguous blocks

CHKDSK

(filename) has invalid cluster, file truncated

CHKDSK

Volume (filename) created (mm, dd, yyyy)

CHKDSK

ASCII Character Codes

DECIMAL VALUE ➡	HEXA DECIMAL VALUE	0	16	32	48	64	80	96	112
⬇		0	1	2	3	4	5	6	7
0	0	BLANK (NULL)	►	BLANK (SPACE)	0	@	P	`	p
1	1	☺	◄	!	1	A	Q	a	q
2	2	☻	↕	"	2	B	R	b	r
3	3	♥	‼	#	3	C	S	c	s
4	4	♦	¶	$	4	D	T	d	t
5	5	♣	§	%	5	E	U	e	u
6	6	♠	▬	&	6	F	V	f	v
7	7	•	↕	'	7	G	W	g	w
8	8	◘	↑	(	8	H	X	h	x
9	9	○	↓	)	9	I	Y	i	y
10	A	◎	→	*	:	J	Z	j	z
11	B	♂	←	+	;	K	[	k	{
12	C	♀	∟	,	<	L	\	l	|
13	D	♪	↔	—	=	M	]	m	}
14	E	♫	▲	.	>	N	^	n	~
15	F	☼	▼	/	?	O	_	o	△

ASCII Character Codes

DECIMAL VALUE →		128	144	160	176	192	208	224	240
↓	HEXA DECIMAL VALUE	8	9	A	B	C	D	E	F
0	0	Ç	É	á	▓	└	╨	∝	≡
1	1	ü	æ	í	▒	┴	╤	β	±
2	2	é	Æ	ó	▓	┬	╥	Γ	≥
3	3	â	ô	ú	│	├	╙	π	≤
4	4	ä	ö	ñ	┤	─	╘	Σ	∫
5	5	à	ò	Ñ	╡	┼	╒	σ	∫
6	6	å	û	ª	╢	╞	╓	μ	÷
7	7	ç	ù	º	╖	╟	╫	τ	≈
8	8	ê	ÿ	¿	╕	╚	╪	Φ	°
9	9	ë	Ö	⌐	╣	╔	┘	Θ	•
10	A	è	Ü	¬	║	╩	┌	Ω	•
11	B	ï	¢	½	╗	╦	█	δ	√
12	C	î	£	¼	╝	╠	▄	∞	n
13	D	ì	¥	¡	╜	═	▌	φ	²
14	E	Ä	₧	«	╛	╬	▐	∈	■
15	F	Å	ƒ	»	┐	╧	▀	∩	BLANK 'FF'

Home of Victor Computers

B

Device I/O Errors

If a disk or device error occurs during a command or program, MS-DOS displays an error message in the following format:

```
<yyy> error <I/O action> <device x>
Abort, Retry, Ignore:
```

In this message, $<$ yyy $>$ can be one of the following:

Write protect error
Bad unit error
Not ready error
Bad command error
Data error
Bad call format error
Seek error
Non-MS-DOS disk error
Sector not found error
No paper error
Write fault error
Read fault error
Disk error

The $<$ I/O-action $>$ can be either of the following:

READING
WRITING

$<$ device x $>$ indicates the device or the drive on which the error has occurred.

"Abort, Retry, Ignore:" is the prompt that MS-DOS provides. MS-DOS waits for you to enter one of the following responses:

▶ **A** for Abort. Abort ends the program requesting the disk read or write and you return to MS-DOS.

▶ **R** for Retry. Retry tells MS-DOS to repeat the operation. Use this response when you have corrected the error. For example, specify Retry if you have closed the drive door or removed a write-protect tab after getting a NOT READY or WRITE PROTECT error.

▶ **I** for Ignore. Ignore tells MS-DOS to ignore the problem and to pretend the error did not occur.

Usually, you should attempt recovery in this order: First specify R to try again; if this is unsuccessful then specify A to end the program. At this point, you should try a new disk or investigate the problem.

You might receive this error message, which could be related to faulty disk read or write:

```
FILE ALLOCATION TABLE BAD FOR DRIVE x
```

This message means that the allocation table copy that resides in memory has pointers to nonexistent blocks. Perhaps your disk was formatted incorrectly, or was not formatted at all. If this error persists, you cannot use the disk. Try using a freshly formatted disk.

C

ANSI Escape Sequences

This appendix explains how the ANSI escape sequences are defined for MS-DOS and gives examples of how to use them. An ANSI escape sequence is a series of characters beginning with an escape character or keystroke that you can use to define functions to MS-DOS. With escape sequences, you can:

▶ Reassign keys

▶ Change graphics functions

▶ Affect cursor movement

Note that the default value is used when you do not specify an explicit value or when you specify zero.

Pn means numeric parameter. Pn is a decimal number specified with ASCII digits. **Ps** means selective parameter. Ps is any decimal number that selects a subfunction. You can select multiple subfunctions by separating the parameters with semicolons.

C.1 Cursor Functions

These escape sequences affect the cursor position on the screen.

CPR—Cursor Position Report (from console driver to system)

ESC [Pn ; Pn R

The CPR sequence reports current cursor position by way of standard input. The first parameter specifies the current line; the second parameter specifies the current column.

CUB—Cursor Backward

ESC [Pn D

This escape sequence moves the cursor back one column without changing lines. The value of Pn determines the number of columns moved. The default value for Pn is 1. The CUB sequence is ignored if the cursor is already in the far left column.

CUD—Cursor Down

ESC [Pn B

This sequence moves the cursor down one line without changing columns. The value of Pn determines the number of lines moved. The default value of Pn is 1. The CUD sequence is ignored if the cursor is already on the bottom line.

CUF—Cursor Forward

ESC [Pn C

The CUF sequence moves the cursor forward one column without changing lines. The value of Pn determines the number of columns moved. The default value for Pn is 1. The CUF sequence is ignored if the cursor is already in the far right column.

CUP—Cursor Position

ESC [Pl ; Pc H

HVP—Horizontal and Vertical Position

ESC [Pl ; Pc f

CUP and HVP move the cursor to the position that you specify with parameters. The first parameter specifies the line number; the second

specifies the column number. The default value is 1. If you do not specify any parameters, the cursor is moved to the home position.

CUU—Cursor Up

ESC [Pn A

This sequence moves the cursor up one line without changing columns. The value of Pn determines the number of lines moved. The default value for Pn is 1. The CUU sequence is ignored if the cursor is already on the top line.

DSR—Device Status Report

ESC [6 n

The console driver outputs a CPR sequence on receipt of the DSR escape sequence.

RCP—Restore Cursor Position

ESC [u

This sequence restores the cursor position to the value it had when the console driver received the SCP sequence (see below).

SCP—Save Cursor Position

ESC [s

The current cursor position is saved. This cursor position can be restored with the RCP sequence.

C.2 Erasing

The following escape sequences affect erase functions.

ED—Erase Display

   ESC [2 J

This sequence erases the screen; the cursor goes to the home position.

EL—Erase Line

   ESC [K

This sequence erases from the cursor position to the end of the line.

C.3 Modes of Operation

The following escape sequences affect screen graphics.

RM—Reset Mode

   ESC [= Ps 1
   ESC [= 1
   ESC [= 0 1
   ESC [? 7 1

Parameters for RM are the same as for SM (Set Mode) except that parameter 7 resets the wrap at the end-of-line mode.

SGR—Set Graphics Rendition

ESC [Ps ; ... ; Ps m

The SGR escape sequence invokes the graphics functions specified by
the parameters described in Table C-1. The graphics functions remain
in effect until the next occurrence of an SGR escape sequence.

Table C-1: Set Graphics Rendition Escape Sequence Parameters

PARAMETER	FUNCTION	
0	All attributes off	
1	Bold on	
4	Underscore on	(monochrome displays only)
5	Blink on	
7	Reverse video on	
8	Concealed on	(ISO 6429 standard)
30	Black foreground	(ISO 6429 standard)
31	Red foreground	(ISO 6429 standard)
32	Green foreground	(ISO 6429 standard)
33	Yellow foreground	(ISO 6429 standard)
34	Blue foreground	(ISO 6429 standard)
35	Magenta foreground	(ISO 6429 standard)
36	Cyan foreground	(ISO 6429 standard)
37	White foreground	(ISO 6429 standard)
40	Black background	(ISO 6429 standard)
41	Red background	(ISO 6429 standard)
42	Green background	(ISO 6429 standard)
43	Yellow background	(ISO 6429 standard)
44	Blue background	(ISO 6429 standard)
45	Magenta background	(ISO 6429 standard)
46	Cyan background	(ISO 6429 standard)
47	White background	(ISO 6429 standard)

C

SM—Set Mode

> **ESC [= Ps h**
>
> **ESC [= h**
>
> **ESC [= O h**
>
> **ESC [? 7 h**

The SM escape sequence changes the screen width or type to one of the parameters described in Table C-2.

Table C-2: Set Mode Escape Sequence Parameters

PARAMETER	FUNCTION
0	40×25 black and white
1	40×25 color
2	80×25 black and white
3	80×25 color
4	320×200 color
5	320×200 black and white
6	640×200 black and white
7	Wrap at end of line

C.4   Keyboard Reassignment

Although not part of the ANSI 3.64-1979 or ISO 6429 standard, the following keyboard reassignments are compatible with these standards. The control sequence is

ESC [Pn ; ... Pn p

or

ESC ["string" ; p

or

ESC [Pn ; "string" ; Pn ; Pn ; "string" ; Pn p

or any other combination of strings and decimal numbers.

The final code in the control sequence (p) is one reserved for private use by the ANSI 3.64-1979 standard. The first ASCII code in the control sequence defines which code is being mapped. The remaining numbers define the sequence of ASCII codes generated when this key is intercepted. There is one exception: if the first code in the sequence is zero (NUL), then the first and second code make up an extended ASCII redefinition.

Examples

1. Reassign the Q and q key to the A and a key (and vice versa):

   ESC [6 5 ; 8 1 p   A becomes Q
   ESC [9 7 ; 1 1 3 p   a becomes q
   ESC [8 1 ; 6 5 p   Q becomes A
   ESC [1 1 3 ; 9 7 p   q becomes a

2. Reassign the F10 key to a DIR command followed by Enter:

   ESC [0 ; 6 8 ; " d i r " ; 1 3 p

   The 0;68 is the extended ASCII code for the F10 key; 13 decimal is the code for Enter.

C.2   Keyboard Management

Also contained in ... the ASCII ... values type the ... display the following ... management ... communication functions.

```
ESC [ p    ... = ...

ESC [ . . . q
```

```
ESC [ m . . . { . . . }
```

The first ... of these ... codes ... are required for the new ... ASCII ... code modifier ... the first ASCII code in the ... The last using ... in ... ASCII code ... the first code is 0, second is ...

Example

1. Here, p ... and the ... with a ... table character ...

```
ESC [ p . . .
ESC [ . . .
[ . . . ]
ESC [ . . .
```

2. ... the ... followed to insert ...

```
ESC [ . . .
```

... is assigned to ...

D

DEBUG

Overview

DEBUG is a debugging program that provides a controlled testing environment for binary and executable object files. DEBUG works on binary files in the same way a text editor works on source files. It lets you alter the contents of a file or CPU register, and then immediately reexecute a program to check the validity of the changes. DEBUG eliminates the need to reassemble a program to see if a problem has been fixed by a minor change.

You can abort all DEBUG commands at any time by pressing Ctrl-C. Ctrl-S freezes the display, so that you can read it before the output scrolls away. Pressing any other key restarts the display.

D.1 Using DEBUG

You can start DEBUG using two methods. With method 1, you type all commands in response to the DEBUG prompt. With method 2, you type all commands at the same time.

Table D-1: Methods to Start DEBUG

METHOD	COMMAND
1	DEBUG
2	DEBUG [filespec [arglist]]

D.1.1 Method 1: Prompts

To start DEBUG using method 1, type

DEBUG ⏎

DEBUG responds with the hyphen (-) prompt, signaling that it is ready to accept your commands. Because you have not specified a filename, you can use other commands to work on current memory, disk sectors, or disk files.

WARNING: When DEBUG starts, it sets up a program header at offset 0 in the program work area. On previous versions of DEBUG you could overwrite this header. You can still overwrite the default header if you do not specify a filespec. If you are debugging a .COM or .EXE file, however, do not tamper with the program header below address 5CH, or DEBUG terminates.

Do not restart a program after the "Program terminated normally" message is displayed. You must reload the program with the N and L commands for it to run correctly.

D.1.2 Method 2: Complete Command Line

To start DEBUG using a command line, type:

DEBUG [filespec [arglist]] ⏎

filespec is the file to be debugged, and arglist is the rest of the command that DEBUG uses when filespec is loaded into memory. arglist is a list of filename parameters and switches that you want passed to the program filespec. You can specify an arglist if you gave a filespec. Thus, when filespec is loaded into memory, it is loaded as if it had been started with the command **filespec arglist**.

If, for example, you type

DEBUG FILE.EXE ↵

DEBUG loads FILE.EXE into memory starting at 100 hexadecimal in the lowest available segment. The BX:CX registers load with the number of bytes placed into memory.

D.2   Commands

Each DEBUG command consists of a single letter followed by one or more parameters. You can use any combination of uppercase and lowercase letters in commands and parameters. **Note:** The control characters and the special editing functions described in this manual apply here.

If you make a syntax error in a DEBUG command, DEBUG reprints the command line and indicates the error with an up-arrow (^) and the word "error." For example:

```
dcs:100 cs:110
   ^ error
```

Table D-2 summarizes DEBUG commands. They are explained in detail, with examples, later in this section.

Table D-2: DEBUG Commands

COMMAND	FUNCTION
A[address]	Assemble
C range address	Compare
D[range]	Dump
E address [list]	Enter
F range list	Fill
G[= address [address...]]	Go
H value value	Hex
I value	Input
L[address [drive record record]]	Load
M range address	Move
N filename filename	Name
O value byte	Output
Q	Quit
R[register-name]	Register
S range list	Search
T[= address][value]	Trace
U[range]	Unassemble
W[address [drive record record]]	Write

All DEBUG commands except Quit accept parameters. You can separate parameters by delimiters (spaces or commas), but you must use a delimiter between two consecutive hexadecimal values. Thus, the following commands are equivalent:

dcs:100 110
d cs:100 110
d,cs:100,110

Table D-3 defines DEBUG command parameters.

Table D-3: Command Parameters

PARAMETER	DEFINITION
drive	A one-digit hexadecimal value that indicates which drive a file is loaded from or written to. These values designate drives as follows: $0 = A:, 1 = B:, 2 = C:, 3 = D:$.
byte	A two-digit hexadecimal value placed in or read from an address or register.
record	A one- to three-digit hexadecimal value that indicates the logical record number on the disk and the number of disk sectors you want to write or load. Logical records correspond to sectors; however, their numbering differs because they represent the entire disk space.
value	A hexadecimal value of up to four digits that specifies a port number or the number of times a command should repeat its functions.
address	A two-part designation consisting of either an alphabetic segment register designation or a four-digit segment address and an offset value. The segment designation or segment address can be omitted; in these cases the default segment is used.
	DS is the default segment for all commands except G, L, T, U, and W. For these commands, the default segment is CS. All numeric values are hexadecimal. In these addresses, for example, you must put a colon between a segment designation (whether numeric or alphabetic) and an offset:
	CS:0100 04BA:0100
range	range consists of two addresses, an L, and a value, where value is the number of lines the command operates on, and L80 is assumed. You cannot use the last form if another hex value follows the range, since the hex value would be interpreted as the second address of the range. For example:
	CS:100 110 CS:100 L 10
	This example is illegal:
	CS:100 CS:110 ^ error
	The limit for range is 10000 hex. To specify a value of 10000 hex within four digits, type 0000 (or 0).

D

PARAMETER	DEFINITION
list	A series of byte values or strings. list must be the last parameter on the command line. For example:
	fcs:100 42 45 52 54 41
string	Any number of characters enclosed in quotation marks. Quotation marks can be single (') or double ("). If the delimiter quotation marks appear within a string, double the quotation marks. For example, the following strings are legal:

'This is a "string" is okay.'

However, this string is illegal:

'This is a 'string' is not.'

Similarly, these strings are legal:

"This is a 'string' is okay."
"This is a ""string"" is okay."

but this string is illegal:

"This is a "string" is not."

Double quotation marks are not needed in the following strings:

"This is a "string" is not necessary."
'This is a ""string"" is not necessary.'

The ASCII values of the characters in the string are used as a list of byte values.

Assemble (A)

A[address]

Assembles 8086/8087/8088 mnemonics directly into memory.

All numeric values are hexadecimal and you must enter them as 1–4 characters. Specify prefix mnemonics in front of the opcode to which they refer. You can also enter 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 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 the defaults with the NEAR or FAR prefix. For 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 you cannot abbreviate the FAR prefix.

DEBUG cannot tell whether some operands refer to a word memory location or to a byte memory location. In these cases, you must explicitly state the data type with the prefix WORD PTR or BYTE PTR. You can also use 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. DEBUG 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
```

Assemble has two popular pseudo-instructions available. The DB opcode assembles byte values directly into memory. The DW opcode assembles word values directly into memory. For example:

```
DB      1,2,3,4,"THIS IS AN EXAMPLE"
DB      'THIS IS A QUOTE: "'
DB      "THIS IS A QUOTE: '"

DW      1000,2000,3000,"BACH"
```

Assemble supports all forms of register indirect commands. For example:

```
ADD     BX,34,[BP+2].[SI-1]
POP     [BP+DI]
PUSH    [SI]
```

Assemble also supports all opcode synonyms. For example:

```
LOOPZ   100
LOOPE   100

JA      200
JNBE    200
```

For 8087 opcodes, you must explicitly specify WAIT or FWAIT. For example:

```
FWAIT FADD ST,ST(3)     ; This line will assemble an
                        ; FWAIT prefix
LD TBYTE PTR [BX]       ; This line will not
```

Compare (C)

C range address

Compares the portion of memory specified by range to a portion of the same size beginning at address.

If the two areas of memory are identical, there is no display and DEBUG returns with the MS-DOS prompt. Differences are displayed in this format:

address1 byte1 byte2 address2

These two commands have the same effect:

C100,200 300

C100L100 300

Each command compares the block of memory from 100 to 1FFH with the block of memory from 300 to 3FFH.

Dump (D)

D[range]

Displays the contents of the specified region of memory.

If you specify a range of addresses, DEBUG displays the contents. If you enter the D command without parameters, 128 bytes display at the first address (DS:100) after the address displayed by the previous Dump command.

The dump displays in two portions: a hexadecimal dump (each byte is shown in hexadecimal value) and an ASCII dump (the bytes are shown in ASCII characters). Nonprinting characters are indicated by a period (.) in the ASCII portion of the display.

The display line shows 16 bytes with a hyphen between the eighth and ninth bytes. Each displayed line begins on a 16-byte boundary.

If you type the command:

dcs:100 110

DEBUG displays:

    04BA:0100 42 45 52 54 41 ... 4E 44 TOM SAWYER

If you type the following command:

D

DEBUG displays 128 bytes. Each line of the display begins with an address, incremented by 16 from the address on the previous line. Each subsequent D (without parameters) displays the bytes immediately following those last displayed.

If you type the command:

DCS:100 L 20

then the display is formatted as described above, but 20H bytes are displayed.

If you then type the command:

DCS:100 115

the display is formatted as described above, but all the bytes in the range from 100H to 115H in the CS segment display.

Enter (E)

E address[list]

Enters byte values into memory at the specified address.

If you type the optional list of values, DEBUG automatically replaces the byte values. If an error occurs, no byte values are changed.

If you type the address without the list, DEBUG displays the address and its contents, then repeats the address on the next line and waits for your input. At this point the Enter command waits for you to do one of the following:

1. Replace a byte value with another value by typing the value after the current value. If the value you type is not a legal hexadecimal value, or if you enter more than two digits, DEBUG does not echo the illegal or extra character.

2. Press the Spacebar to advance to the next byte. To change the value, enter the new value after the current value. If you space beyond an 8-byte boundary, DEBUG starts a new display line with the address displayed at the beginning.

3. Type a hyphen (-) to return to the preceding byte. If you decide to change a byte behind the current position, type the hyphen to return the current position to the previous byte. When you type the hyphen, a new line is started with the address and its byte value displayed.

4. Press the Enter key to terminate the Enter command. You can press the Enter key at any byte position.

Assume you enter the following command:

ECS:100

DEBUG displays

```
04BA:0100  EB._
```

To change this value to 41, type 41 as shown:

04BA:0100 EB.41_

To step through the subsequent bytes, press the Spacebar to see

```
04BA:0100  EB.41    10.    00.    BC._
```

To change BC to 42, type

04BA:0100 EB.41   10.   00.   BC.42_

Now, to change 10 to 6F, type the hyphen as many times as needed to return to byte 0101 (value 10). Then replace 10 with 6F:

```
04BA:0100  EB.41    10.    00.    BC.42-
04BA:0102  00.-_
04BA:0101  10.6F_
```

Press Enter to end the Enter command and return to the DEBUG command level.

Fill (F)

F range list

Fills the addresses in the range with values in the list.

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.

Assume you type the following command:

F04BA:100 L 100 42 45 52 54 41

DEBUG fills memory locations 04BA:100 through 04BA:1FF with the bytes specified. The five values are repeated until all 100H bytes are filled.

Go (G)

G[= address[address...]]

Executes the program currently in memory.

If you type only the Go command, the program runs as it would outside DEBUG.

If you set = address, execution begins at the address specified. The equals sign (=) is required, so that DEBUG can distinguish the start = address from the breakpoint addresses.

When the other optional addresses are set, execution stops at the first address encountered, regardless of that address's position in the list of addresses to halt execution or program branching. When program execution reaches a breakpoint, the registers, flags, and decoded instruction display for the last instruction executed. The result is the same as if you had entered the Register command for the breakpoint address.

You can set up to ten breakpoints. Breakpoints must be set, however, only at addresses containing the first byte of an 8086-88 opcode. If you set more than ten breakpoints, DEBUG returns the BP error message.

The user stack pointer must be valid and have 6 bytes available for this command. The Go 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. Thus, if the user stack is not valid or is too small, MS-DOS can crash. DEBUG places an interrupt code (0CCH) at the specified breakpoint address(es).

When DEBUG encounters an instruction containing the breakpoint code, all breakpoint addresses are restored to their original instructions. If execution does not halt at one of the breakpoints, the interrupt codes are not replaced with the original instructions.

Assume 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, and the Go command terminates.

After DEBUG encounters a breakpoint, you can type the Go command again and the program executes just 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.

D-14

Hex (H)

H value value

Performs hexadecimal arithmetic on the two parameters specified.

DEBUG adds the two parameters and subtracts the second parameter from the first. The results of the arithmetic display on a single line: first the sum, then the difference.

If you type the command:

H19F 10A

DEBUG performs the calculations and then displays the results:

    02A9 0095

Input (I)

I value

Inputs and displays one byte from the port specified by value.

This command allows a 16-bit port address.

Assume you type the following command:

I2F8

Assume also that the byte at the port is 42H. DEBUG inputs the byte and displays the value 42.

Load (L)

L[address [drive record record]]

Loads a file into memory.

Set BX:CX to the number of bytes read. The file loaded must already be named. Name the file either when you start DEBUG or with the N 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 type the L 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 address specified.

If you type L with all the parameters, DEBUG loads absolute disk sectors instead of a file. The records are taken from the drive specified. The drive designation is numeric—0 = A:, 1 = B:, 2 = C:. DEBUG begins loading with the first record specified, and continues until the number of sectors specified in the second record are loaded.

Assume you type the following commands:

```
A > DEBUG
-NFILE.COM
```

To load FILE.COM, type L. DEBUG loads the file and displays the DEBUG prompt. To load portions of a file or certain records from a disk, type

```
L04BA:100 2 0F 6D
```

DEBUG then begins with logical record number 15 and loads 109 (6D hex) records into memory beginning at address 04BA:0100. When the records are loaded, 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. Thus, the size of an .EXE file on disk differs from its size in memory.

If the named file is a .HEX file, typing the L command with no parameters tells DEBUG to load the file beginning at the address specified in the .HEX file. If the L command includes the option address, DEBUG determines the start address by adding the address specified in the L command to the address found in the .HEX file.

Move (M)

M range address

Moves the block of memory specified by range to the location beginning at the address specified.

DEBUG always performs overlapping moves. Overlapping moves are where part of the block overlaps some of the current addresses without loss of data. Addresses that could be overwritten are moved first. When you want to move from higher addresses to lower addresses, this command moves the data beginning at the block's lowest address and then works toward the highest. When you want to move from lower addresses to higher addresses, DEBUG moves the data beginning at the block's highest address and works toward the lowest.

The M command actually copies data rather than moves it. If you do not plan to write new data to the addresses in the block you are moving, the existing data remains intact. Consequently, the sequence of the move is important.

Assume that you type

MCS:100 110 CS:500

DEBUG first moves data from address CS:110 to address CS:510, then CS:10F to CS:50F, and so on until CS:100 is moved to CS:500. You can review the results of the move by typing the D command, with the same address you typed for the M command.

Name (N)

N filename [filename]

Sets filenames.

The Name command performs two functions:

1. Name assigns a filename for a later Load or Write command. Thus, if you start DEBUG without naming a file to debug, you must type the N filename command before a file can be loaded.

2. Name assigns filename parameters to the file you are debugging. In this case, Name accepts a list of parameters that are used by the file being debugged.

These two functions overlap. Consider the following set of DEBUG commands:

-NFILE1.EXE
-L
-G

These commands result in four steps:

1. (N)ame assigns the filename FILE1.EXE to the filename used in any later Load or Write commands.

2. (N)ame also assigns the filename FILE1.EXE to the first filename parameter used by any program that is later debugged.

3. (L)oad loads FILE1.EXE into memory.

4. (G)o executes FILE1.EXE with FILE1.EXE as the single filename parameter, that is, FILE1.EXE is executed as if FILE1.EXE had been typed at the command level.

A more useful chain of commands might be:

```
-NFILE1.EXE
-L
-NFILE2.DAT FILE3.DAT
-G
```

Here, Name sets FILE1.EXE as the filename for the subsequent Load command. The Load command loads FILE1.EXE into memory, and then the Name command is used again, this time to specify the parameters used by FILE1.EXE. Finally, when the Go command is executed, FILE1.EXE executes as if you had typed FILE1 FILE2.DAT FILE3.DAT at the MS-DOS command level.

If DEBUG executes a Write command, then FILE1.EXE—the file being debugged—is saved with the name FILE2.DAT. To avoid these results, always execute a Name command before either a Load or a Write command.

The Name command can affect four regions of memory:

CS:5C   FCB for file 1

CS:6C   FCB for file 2

CS:80   Count of characters

CS:81   All characters typed

DEBUG sets up a File Control Block (FCB) for the first filename parameter you gave at CS:5C. If you type a second filename parameter, you set up an FCB beginning at CS:6C. The number of characters you type, exclusive of the first character, "N", is given at location CS:80. The actual stream of characters given by the command (again, exclusive of the letter "N") begins at CS:81.

This stream of characters might contain switches and delimiters that would be legal in any command typed at the MS-DOS command level.

A typical use of the Name command is

DEBUG PROG.COM
-NPARAM1 PARAM2/C
-G
-

In this example, the Go command executes the file in memory as if you had typed the following command line:

PROG PARAM1 PARAM2/C

Output (O)

O value byte

Send the byte specified to the output port specified by value.

This command allows a 16-bit port address.

If you type **O2F8 4F**, DEBUG outputs the byte value 4F to output port 2F8.

Quit (Q)

Q

Terminates the DEBUG utility.

The Q command takes no parameters and exits DEBUG without saving the file you are currently working on. You are returned to the MS-DOS command level.

To end the debugging session, type

Q

DEBUG terminates and control returns to the MS-DOS command level.

Register (R)

R[register-name]

Displays the contents of one or more CPU registers.

If you do not type register-name, the R command dumps the register save area and displays the contents of all registers and flags.

If you type a register name, the 16-bit value of that register displays in hexadecimal and a colon appears as a prompt. You can then either type a value to change the register, or, if you do not want any changes, press Enter.

The only valid register names are

AX	DI	PC
BP	DX	SI
BX	ES	SP
CS	F	SS
CX	IP	SX

(IP and PC both refer to the Instruction Pointer.)

Any other entry for register-name results in a BR error message. See the end of this appendix for a list of DEBUG error messages.

If you enter F as the register-name, DEBUG displays each flag with a two-character alphabetic code. To alter any flag, type the opposite two-letter code. The flags are either set or cleared.

Table D-4 lists the flags with their codes for SET and CLEAR.

Table D-4: Register Flag Codes

FLAG NAME	SET CODE	CLEAR CODE
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

Whenever you type the RF command, the flags display in a row at the beginning of a line, in the order shown in Table D-4. At the end of the list of flags, DEBUG displays a hyphen (-). You can enter new flag values as alphabetic pairs, in any order. You do not have to leave spaces between the flag entries.

To exit the R command, press the Enter key. Any flags for which you did not enter new values remain unchanged.

If you enter more than one value for a flag, DEBUG returns a DF error message. If you enter a flag code other than those shown in Table D-4, DEBUG returns a BF error message. In both cases, flags up to the error in the list are changed; those flags at and after the error are not.

At startup, 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.

If you type

R

DEBUG displays all registers, flags, and the decoded instruction for the current location. If the location is CS:11A, the display looks 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 type

RF

DEBUG displays the flags:

```
NV UP DI NG NZ AC PE NC - _
```

Now type any valid flag designation, in any order, with or without spaces. For example:

NV UP DI NG NZ AC PE NC - PLEICY

DEBUG responds only with the DEBUG prompt. To see the changes, type either the R or RF command:

RF

DEBUG displays

```
NV UP EI PL NZ AC PE CY - _
```

Press Enter to leave the flags this way, or to specify different flag values.

Search (S)

S range list

Searches the range specified for the list of bytes specified.

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.

If you type

SCS:100 110 41

DEBUG responds

```
04BA:0104
04BA:010D
-type:
```

Trace (T)

T[= address][value]

Executes one instruction and displays the contents of the decoded instruction and all registers and flags.

If you type the optional = address, DEBUG traces at the address specified. The optional value tells 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.

If you type

T

DEBUG returns a display of the registers, flags, and decoded instruction for that one instruction. Assume that your current position is 04BA:011A; DEBUG might return the 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

T = 011A 10

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 until the last instruction is executed. Then the display stops, and you can see the register and flag values for the last few instructions performed. Remember that Ctrl-S suspends the display at any time, so that you can study the registers and flags for any instruction.

Unassemble (U)

Disassembles bytes and displays the source statements that correspond to them, with addresses and byte values.

The display of disassembled code looks like a listing for an assembled file. If you type the U command without parameters, DEBUG disassembles 20 hexadecimal bytes at the first address after that displayed by the previous Unassemble command. If you type the U command with the range parameter, then DEBUG disassembles all bytes in the range. If the range is given as an address only, then 20H bytes are disassembled instead of 80H that the Dump command would default to.

If you type

U04BA:100 L10

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     66
04BA:010D    69        DB     69
04BA:010E    63        DB     63
04BA:010F    61        DB     61
```

If you enter

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 you change bytes in some addresses, the disassembler alters the in-
struction statements. You can type the U command for the changed
locations, the new instructions viewed, and the disassembled code used
to edit the source file.

Write (W)

W[address[drive record record]]

Writes the file being debugged to a disk file.

If you type W with no parameters, 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, DEBUG
writes the file beginning at that address.

If you use a G or T command, BX:CX must be reset before you use
the Write command without parameters. Note that if DEBUG loads
and modifies a file, the name, length, and starting address are all set
correctly to save the modified file, as long as the length has not
changed.

The file must be named either with the DEBUG invocation command
or with the N 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 W command with parameters, DEBUG writes the file beginning from the memory address specified. The file is written to the specified drive (the drive designation is numeric here—0 = A, 1 = B, 2 = C, and so on) beginning at the logical record number specified by the first record. DEBUG continues until the number of sectors specified in the second record are written.

Note: Writing to absolute sectors is extremely dangerous because the process bypasses the file handler.

If you type

   W

DEBUG writes the file to disk and then displays the DEBUG prompt.

If you type

   W
   CS:100 1 37 2B

DEBUG writes out the contents of memory, beginning with the address CS:100, to the disk in drive B:. The data written out starts at logical record number 37H and consists of 2BH records. When the write is finished, DEBUG displays

```
WCS:100 1 37 2B
_
```

D.3 DEBUG Error Messages

You might see any of the following error messages during a DEBUG session. Each error terminates DEBUG command under which it occurs, but it does not terminate DEBUG itself.

ERROR CODE	DEFINITION
BF	Bad flag: You attempted to alter a flag, but the characters typed were not one of the acceptable pairs of flag values. See the Register command for the list of acceptable flag entries.
BP	Too many breakpoints: You specified more than ten breakpoints as parameters to the Go command. Retype the Go command with ten or fewer breakpoints.
BR	Bad register: You typed the R command with an invalid register name. See the Register 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.

Index