NEC

MS-DOS® 3.3
User's Guide

NEC
NEC Corporation

819-180039-INT Rev. 01
5-88

Contents

Page

Section 2 Getting Acquainted with MS–DOS

Contents

Page

Section 3 More About Files and Directories

Section 4 MS–DOS Commands

Section 5 Batch Processing

Section 6 MS–DOS Editing and Function Keys

Contents

Page

Section 7 The Line Editor (EDLIN)

Section 8 EDLIN Commands

Section 9 File Comparison Utility (FC)

Section 10 LINK: A Linker

Contents

Page

Section 11 DEBUG

Appendix A System Setup Utility

Contents

Page

Appendix B Keyboards

Appendix C How to Configure Your System

Appendix D Installable Device Drivers

Contents

Figures

Tables

Preface

The *MS–DOS 3.3 User's Guide* describes how to set up and use the MS–DOS® operating system on your PowerMate™.

Section 1

Introduction

The *MS-DOS 3.3 User's Guide* provides the information you need to use version 3.3 of the Microsoft® MS-DOS operating system on your NEC PowerMate personal computer. Among the topics covered in this guide are

- instructions for setting up MS-DOS
- a brief introduction to using MS-DOS
- detailed reference material on MS-DOS commands
- descriptions of MS-DOS editing and programming tools such as EDLIN, DEBUG, FC, and LINK
- listings of MS-DOS error messages and what to do when you see them.

Before you can use your computer to run application programs or use any feature of MS-DOS, you must follow the System Setup instructions in Appendix A of this guide. When you have completed System Setup, read Section 2, "Getting Acquainted with MS-DOS," if you need an introduction to MS-DOS. It explains the terms you'll need to know, and describes how to perform everyday tasks such as copying files, formatting diskettes, and issuing commands.

The remaining sections in this guide cover specific aspects of the MS-DOS operating system as outlined below.

Section 3 goes into detail about using files and directories in MS-DOS.

Section 4 lists the MS-DOS commands in alphabetical order and gives detailed descriptions, comments, and examples.

Section 5 describes how to make a batch file.

Section 6 tells you how to use MS-DOS editing and function keys.

Section 7 describes the use of EDLIN, the line editor program.

Section 8 goes into detail about EDLIN commands.

Section 9 describes the use of FC, the file comparison utility.

Section 10 explains how to use the Microsoft 8086 Object Linker, LINK, to create executable programs.

Section 11 explains how to use DEBUG, a debugging utility.

Appendix A describes the System Setup utility. Follow the instructions in Appendix A to install MS-DOS.

Appendix B shows the non-U.S. keyboard layouts and describes use of the Alt key and special key combinations.

Appendix C explains how to create a configuration file.

Appendix D describes installable device drivers.

Appendix E explains how to use code pages.

Appendix F lists disk and device error messages.

Appendix G lists the MS-DOS message directory.

Appendix H describes how to use FDISK, the hard disk partition control utility.

Section 2

Getting Acquainted with MS-DOS

WHAT IS MS-DOS?

The Microsoft disk operating system (MS-DOS) is like a translator between you and your computer. Basically, this translator is a series of programs, which let you communicate with your computer, your disk drives, and your printer, so that you can manage these resources to your advantage.

You can use MS-DOS programs on computers with floppy disks or hard disks. Once you have loaded MS-DOS into your computer's memory, you can compose letters and reports, run programs and languages such as Microsoft GW-BASIC®, and use devices such as printers and disk drives.

WHAT'S NEW IN MS-DOS VERSION 3.3?

This version of MS-DOS has several new features, including

- Performance improvements. Several internal enhancements and a new command, FASTOPEN, improve MS-DOS performance, especially with applications that use many files.
- Support for 1.44 megabyte, 3 1/2-inch disks.
- Support for four serial input/output ports. Previous versions of MS-DOS supported two.
- Enhancements to CONFIG.SYS commands including BUFFERS, STACKS, and DEVICE. MS-DOS 3.3 also includes a new CONFIG.SYS command, COUNTRY.
- Ability to use partitions larger than 32 megabytes. You can create extended DOS partitions for disks larger than 32 megabytes with FDISK.

- Improved national language support. MS–DOS 3.3 offers the ability to select alternate language–specific character sets (called code pages) instead of the standard U.S. character set. Three new commands — CHCP, NLSFUNC, and SELECT — and several enhanced MS–DOS commands (including KEYB and MODE) support code page selection. In addition the CONFIG.SYS command, COUNTRY, and two new installable device drivers allow you to select language–specific code pages.
- Enhanced batch processing capabilities.
- Ability to make a system disk formatted by an earlier version of MS–DOS bootable with MS–DOS 3.3 allows the IO.SYS system file to be noncontiguous on the disk.
- Various enhancements to existing commands, including APPEND, FORMAT, MODE, ATTRIB, GRAPHICS, RESTORE, BACKUP, KEYB, TIME, and DATE.

TERMS YOU SHOULD KNOW

When you are introduced to a new or different idea, you must often learn a new set of words to understand the idea. MS–DOS is no exception. The following pages explain some terms you will need to know so that you can read and use this manual.

Program

Programs, often called application programs, applications, or software, are series of instructions written in computer languages. These instructions are stored in files and tell your computer to perform a task. For example, a program might tell your computer to alphabetically sort a list of names.

File

A file is a collection of related information, like the contents of a file folder in a desk drawer. File folders, for instance, might contain business letters, office memos, or monthly sales data. Files on your disks could also contain letters, memos, or data.

Filename

Just as each folder in a file cabinet has a label, each file on a disk has a name. This name has two parts: a filename and an extension. A filename can be from 1 to 8 characters in length, and can be typed in uppercase or lowercase letters. MS–DOS automatically converts filenames to uppercase letters. In this manual, filenames and extensions are shown in upper case.

Filename extensions consist of a period followed by one, two, or three characters. Extensions are optional, but it's a good idea to use them, since they are useful for describing the contents of a file to you and to MS–DOS. For 'nstance, if you want to be able to quickly identify your report files, you can add the file name extension .RPT to each one. Here's an example of a file name with this extension:

```
PROGRESS.RPT
```

Directory

A directory is a table of contents for a disk. It contains the names of your files, their sizes, and the dates they were last modified.

When you look at the directory on your MS–DOS master diskette, you will see many files with the extension .EXE or .COM. The extension .EXE means executable, and .COM means command. These extensions tell MS–DOS that the files are programs that can be run. Many files will have other kinds of extensions, such as .DOC and .TXT, which might contain text. Another common program file extension is .BAS for BASIC programs.

Volume Label

When you use a new disk, you can put a label on the outside of it to help you identify its contents. You can also give each of your disks an internal name, called a *volume label.*

You can look at the volume label on a disk by displaying its directory. Some programs may look at the volume label to see if you are using the correct disk. So make sure that you label your disks.

Disk Drive

Diskette drives are commonly referred to as the A drive and the B drive; a hard disk is usually referred to as the C drive. Check your *PowerMate Owner's Guide* to see which drive is A and which is B (or C).

Drive Name

A complete drive name consists of a drive letter and a colon. When using a command, you may need to type a drive name before your file name to tell MS–DOS where to find the disk that contains your file. For example, suppose you have a file named FINANCES.DOC on the disk in drive B. To tell MS–DOS where to find this file you would type the drive name before the file name:

```
B:FINANCES.DOC
```

If you don't specify a drive name when you type a file name, MS-DOS automatically searches for the file on the disk in the default drive. To let you know that it is ready to receive a command, MS-DOS displays a prompt that contains the default drive letter followed by a greater-than sign. Following the prompt on your screen is the cursor, the small box or flashing underline that shows where the next character you type will appear.

So when your prompt is A>, MS-DOS searches only the diskette in drive A for files and programs--unless you tell it to search in another drive.

If you will be working primarily with files on drive B, it is easier to change the default drive to B, so that you won't have to type the letter B, followed by a colon, with every command and file name.

Command

Just as you will run programs to create and update files containing your data, you will also need to run some special programs, called MS-DOS commands, that let you manipulate entire files.

When you give MS-DOS certain commands, you are asking the computer to perform tasks. For example, when you use the DISKCOPY command to copy your MS-DOS master disk, you are running a program named DISKCOPY on the MS-DOS disk.

Other MS-DOS commands perform the following functions:

- compare, copy, display, delete, and rename files
- copy, format, and label disks
- run your programs, as well as those supplied with MS-DOS, such as EDLIN
- list directories
- enter the date and time
- set printer and screen options.

Commands appear in uppercase letters in this manual.

THE MS–DOS KEYBOARD

Your computer keyboard (shown in Figures 2–1—2–3) has some keys that have special meanings to MS–DOS.

The space bar moves the cursor to the right and erases the characters.

Cursor control keys move the cursor right, left, up, and down. They do not affect the characters that are displayed. Some programs ignore these keys or do not use them. In this manual, the direction keys are also referred to as the right, left, up, and down arrow keys.

Back Space moves the cursor to the left one character at a time and erases characters it moves over. Use Back Space to correct typing errors on the current line.

Enter (↵) sends a command to MS–DOS. Press this key after you have typed a command line.

Pause causes the system to pause and wait for any key (except Pause) to be pressed to resume operation. This provides a method of temporarily suspending list, print, and other output functions.

Print Screen causes the system to print out everything currently displayed on the screen (text data or graphics). Note that the GRAPHICS.COM or CRTDUMP.COM program must be run before a graphics screen can be printed. Refer to the GRAPHICS and CRTDUMP commands in Section 4.

SHIFT KEY PRIORITIES AND COMBINATIONS

Shift shifts keys to the uppercase character. If you press Caps Lock and the Caps Lock indicator lights, pressing Shift invokes the unshifted state for alphabetic characters. Shift also temporarily reverses the Num Lock, or non–lock state of the numeric keypad keys.

Num Lock shifts the numeric keypad keys into a "numbers only" state. Pressing Num Lock a second time releases the keyboard from the Num Lock state so that the cursor control functions become active.

A small green indicator labeled "Num Lock" lights when the Num Lock state is in effect.

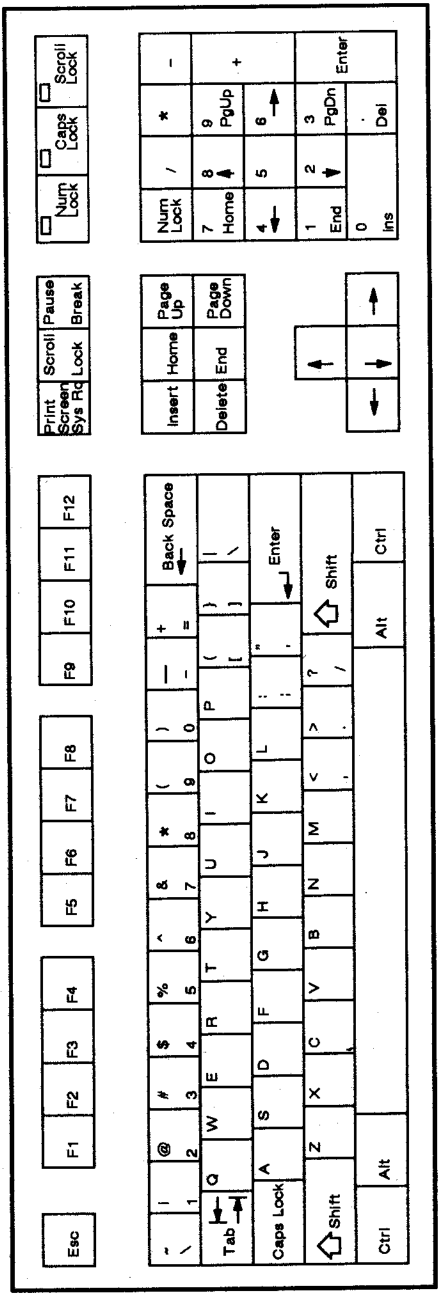


Figure 2–1 101–Key Keyboard

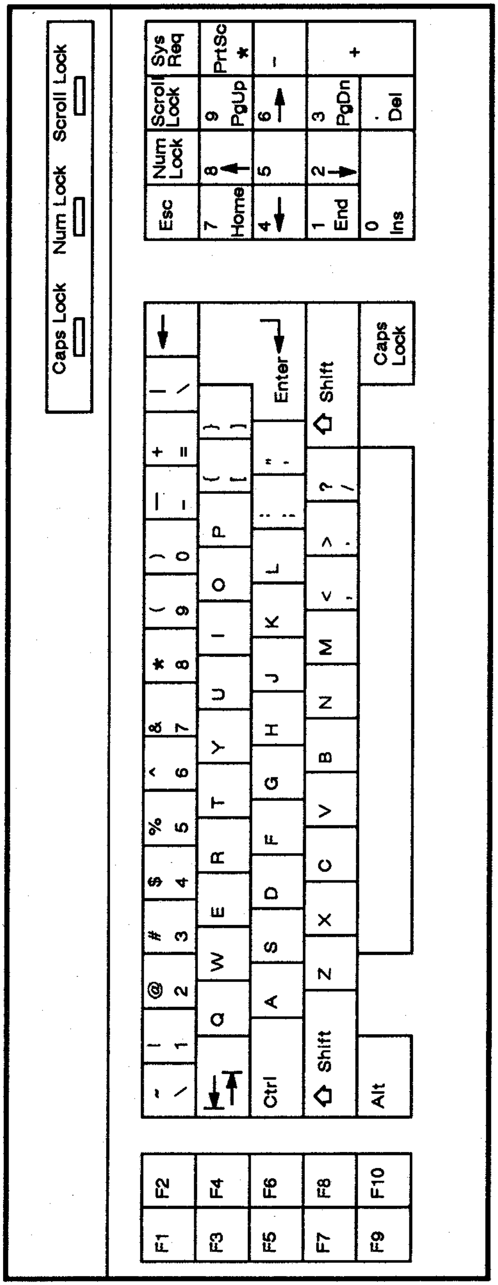


Figure 2–2 84–Key Keyboard

The control key (Ctrl) has a special task. It lets you give complex commands to your computer by pressing only two or three keys. You must hold down the Ctrl key while you press another key. That is, you use the Ctrl key as you would the Shift key.

When you press the Ctrl key and the S key at the same time, you can stop the scrolling of the screen display. Then to continue scrolling, press Ctrl–S again.

When you press the Ctrl key and the C key at the same time, you can stop a command.

If you want to start (or restart) MS–DOS, press the Ctrl, Alt, and Del keys at the same time.

Caps Lock shifts alphabetic character keys to uppercase. Pressing the Caps Lock key a second time reverses the action. A small green indicator labeled "Caps Lock" lights up when the Caps Lock state is in effect.

Scroll Lock is interpreted by some application programs as indicating that the use of the cursor control keys should cause windowing over the text rather than cursor movement. Pressing the Scroll Lock key a second time reverses this action. The keyboard routine simply records the current shift state of the Scroll Lock key. It is the responsibility of the application program to perform the function.

A small green indicator labeled "Scroll Lock" lights up when the Scroll Lock state is in effect.

DISKETTES AND HARD DISKS

A diskette is a magnetized disk that can store hundreds of single–spaced pages of text. Every diskette is enclosed in its own protective cover. The front of this cover is smooth, while the back has visible seams. You should always place labels on the front of the cover, at the top, so that the label doesn't touch the magnetic surface of the disk. It's also a good idea to use a felt–tip pen when writing on labels––a pencil or ballpoint pen can damage the disk if you press too hard.

You should store diskettes in a safe place, away from dust, moisture, magnetism, and extreme temperatures. Be sure to label each diskette you use, since labels help you identify what files are on the disk and remind you that the diskette has information stored on it.

A microdiskette is a 3 1/2–inch version of a diskette contained in a hard plastic jacket. The magnetic surface of the disk is protected by an automatic shutter which is retracted only when the microdiskette is inserted in the drive. Although sturdier than standard diskettes, microdiskettes should also be handled carefully and stored in a safe place.

Diskette Protection

Labels do help you keep track of your diskettes, but you may also need to protect your diskettes. Some diskettes are protected, letting you examine information on them without letting you change anything. These are called *write–protected diskettes.*

For 5 1/4–inch diskettes, there is usually a small piece of tape, called a tab, covering a notch on the right side of the diskette.

Microdiskettes have a sliding write–protect tab. When the tab is slid to expose the small hole, the microdiskette is write–protected.

You can copy information onto a write–protected diskette by removing or sliding the write–protect tab; however, you should consider why the diskette was protected before you change its contents. After you have copied or changed a write–protected diskette it's always a good idea to replace the write–protect tab.

If a 5 1/4–inch diskette does not have a write–protect notch, it is always write–protected. Many application programs, including this version of MS–DOS, come on write–protected diskettes that protect the files from being destroyed accidentally.

Hard Disks

In addition to diskettes, some computers use a hard disk, which can store much more information than a diskette. Computers also take less time to find information stored on a hard disk than on a diskette. A hard disk is usually built into the computer.

When you store application programs, including MS–DOS, on your hard disk, you should keep a backup copy of the programs on a diskette in case the information on the hard disk is accidentally damaged or destroyed.

The FORMAT Command

Before you can use your new diskettes for storing information, you must format them. You do this with the FORMAT command, a special program that structures a diskette so that MS–DOS can find information on it. The FORMAT command also checks the disk for defective spots.

You can format diskettes, microdiskettes, and hard disks. But remember that if a disk is not blank, formatting it destroys any data already on the disk.

HOW TO NAME YOUR FILES

When naming a file you may have trouble finding a name that uniquely identifies the file's contents. Dates, for example, are often used in file names; however, they take up several characters, leaving you with little flexibility. Other common names for files are words like BUDGET, FINANCES, ANALYSIS, and REPORT. These kinds of file names identify the contents, but leave little room for dates. So the secret is to find a compromise--a point where you can combine a date with a word, creating a unique file name.

The name of a typical MS-DOS file looks like this:

CUSTOMER.LST

In this manual, file names appear in uppercase letters. You can type file names in uppercase or lowercase letters, even though MS-DOS converts them into uppercase letters. Some more examples of file names are:

BUDGET.86
TAKEOVER.BID
JUNE86
FINANCES.DOC
SCHEDULE.MAY

Many of your file names will contain only letters and numbers. But you may also use any of the following symbols (and letters) in your file names and extensions:

A-Z a-z 0-9 $ % ’ - @ { } ~ ‘ ! # ( ) &

Some applications may not let you use all of these symbols. If in doubt use only letters and numbers.

Although you do have some freedom when naming your files, there are certain names that you may not use, because MS-DOS reserves them for specific devices that your computer uses. These invalid names are: AUX, CLOCK$, COM, CON, LPT, LST, NUL, and PRN. You may use these names as extensions, but remember not to use them to name your files.

DIRECTORIES

The names of your files are kept in a directory on each disk. The directory also contains information on the sizes of the files, and the dates they were created and updated.

If you want to know what files are on your disk, you can use the DIR command. This command tells MS-DOS to display all the files in a specific directory on a disk.

You can also get information about any file on your disk by typing the DIR command followed by a file name. For example, to display directory information for a file named SCHEDULE, you could use the following command:

```
DIR SCHEDULE
```

MS-DOS would respond by displaying the file name SCHEDULE followed by the file's size in bytes and the date and time it was last changed.

HOW TO START MS-DOS

To start MS-DOS from a diskette, make sure your PowerMate is turned off. Insert the copy of the MS-DOS System diskette (the one you made with the System Setup utility) in drive A and power-on the PowerMate.

To start MS-DOS from a hard disk, just power-on the PowerMate. MS-DOS will start automatically.

HOW TO QUIT

There is no "quit" command in MS-DOS, but you can end your MS-DOS session easily by following these steps:

1. Make sure that your last command is finished. You should see the MS-DOS prompt (usually A> or C>) on the screen.
2. Remove the diskettes from the drives and store them in a safe place.
3. Turn off your PowerMate.
4. Turn off your monitor.

FILE COMMANDS

You can use several MS–DOS commands to manage your files. Some of the more common commands are DIR, COPY, DEL, RENAME, and PRINT.

The following examples assume that drive A is the default drive.

The DIR Command

If you want to find out what files are on a disk, you can list its directory by using the MS–DOS DIR command. For example, to display the directory of the disk in drive B, you would use the following command:

```
DIR B:
```

If you use the DIR command without a drive letter, MS–DOS lists the directory of the disk in the default drive.

The COPY Command

If you need to copy files, you can use the COPY command to copy one or more files, either on the same disk or from one disk to another. For instance, suppose you need a copy of a file named SALES.DOC that you have on a diskette in drive A, and suppose you want to call this new copy MONTHLY.RPT:

To copy the SALES.DOC file and call the new copy MONTHLY.RPT you would just follow these steps:

1. Make sure that the diskette with the SALES.DOC file is in drive A and that A is the default drive.
2. At the MS–DOS prompt, type the following command:

   ```
   COPY SALES.DOC MONTHLY.RPT
   ```

3. Press **Enter**.

When you make a copy of a file, you cannot give the new file the same name as the original file. You can, however, copy a file from one disk to another and keep the same file name. For example, to copy a file from the diskette in drive A to the diskette in drive B, use the following command:

```
COPY A:SALES.DOC B:SALES.DOC
```

In the above example, if A is the default drive (that is, if the prompt is A>), you needn't type the letter A, followed by a colon, before the first file name. Also, by default, the copy will have the name of the original file if you do not specify a new name. For example, the following commands all produce the same result:

```
COPY A:SALES.DOC B:SALES.DOC
COPY SALES.DOC B:SALES.DOC
COPY SALES.DOC B:
```

The DEL Command

Just as you may need to make copies of files, you may also need to remove old or unnecessary files to clean up your file system. So when you want to erase a file from a disk, you can use the MS-DOS DEL command. Remember, though, that the DEL command permanently erases the file. To delete an old SALES.DOC file from the disk in drive B, at the MS-DOS prompt you would use the following command:

```
DEL B:SALES.DOC
```

Suppose you have an old copy of the SALES.DOC file that you no longer need. To delete this file from the disk in the default drive, you would just follow these steps:

1. Make sure that the disk with the SALES.DOC file is in the default drive.
2. At the MS-DOS prompt, type the following command:

   ```
   DEL SALES.DOC
   ```
3. Press **Enter**. MS-DOS then deletes the SALES.DOC file from the disk.

To MS-DOS, the wildcard name *.* means all files in a directory, so be careful when you use this abbreviation in your commands. For example, if you type the DEL command, followed by *.*, MS-DOS asks if you want to delete all the files in the directory; if you then press Y, (for Yes) MS-DOS permanently deletes the files in the directory.

The DEL command does not work if you type the word "DELETE." You can, however, type ERASE instead of DEL.

The RENAME Command

Occasionally, you may want to change the name of a file. For example, suppose you have a file named MONTHLY.RPT on a disk. When you add other monthly reports to your disk, you may want to change the name to something more specific. To change the name to ANNUAL.RPT, for instance, you would use the following command:

```
RENAME MONTHLY.RPT ANNUAL.RPT
```

You can only rename files on the same disk, so you cannot change A:MONTHLY.RPT to B:MONTHLY.RPT.

Suppose you want to rename a file named PAYROLL.DOC, on the disk in the default drive, to SALARY.DOC. You would simply follow these steps:

1. Make sure that the disk with the PAYROLL.DOC file is in the default drive.
2. At the MS–DOS prompt, type the following command:

   ```
   RENAME PAYROLL.DOC SALARY.DOC
   ```
3. Press **Enter**.

The RENAME command can be abbreviated to REN.

The TYPE Command

If you want MS–DOS to display a file that contains text (often called a text file) on the screen, use the TYPE command. For example, say you have created a file named PHONE.LST on the disk in drive A, and you want to check one of the phone numbers. To display the file on the screen, you would use the following command:

```
TYPE A:PHONE.LST
```

Suppose you want to check your employees' salary figures. So you decide to look at a file named SALARY.DOC that is on the disk in the default drive. To display the SALARY.DOC file you would just follow these steps:

1. Make sure that the disk with the SALARY.DOC file is in the default drive.
2. At the MS–DOS prompt, type the following command:

   ```
   TYPE SALARY.DOC
   ```
3. Press the **Enter** key.

MS–DOS then displays the SALARY.DOC file on the screen.

If the file is too long to fit on the screen, remember that you can press Ctrl–S to prevent it from scrolling off the screen. When you press Ctrl–S again, the file will resume scrolling.

MS–DOS displays only text files on the screen. So if you try to display a program file (one with an extension of .COM or .EXE), you will see only strange symbols on the screen.

If you have an application program that creates files, you may need to run the application to view them.

The PRINT Command

If you have a printer attached to your computer, you can print files with the MS–DOS print command. Assume, for example, that you have a file named INVEST.MNT and want to print it on your printer. You could use the following command:

```
PRINT INVEST.MNT
```

Say you have a file that contains a list of investors and their phone numbers, and suppose you want to print this file and keep it near your phone. The file is named INVEST.MNT and is on the disk in drive B. Drive A is the default drive (A> is the prompt). To print the INVEST.MNT file, you would just follow these steps:

1. Make sure that the MS–DOS disk is in drive A.
2. Make sure that the disk with the INVEST.MNT file is in drive B.
3. Check to see that your printer is on and has paper.
4. At the MS–DOS prompt, type the following command:

   ```
   PRINT B:INVEST.MNT
   ```
5. Press the **Enter** key.
6. MS–DOS prompts you for the name of the printing device connected to your computer. (This name is usually the communications port that the printer cable connects to.) Just type the name, or press **Enter** to print to the default printer.

If the master diskette is not in drive A, MS–DOS prompts you to insert it in the drive.

While a file is being printed, you can type other commands to MS–DOS. You can even run other programs or create and modify files. But since printing a file takes a lot of your computer's resources, your tasks may take longer if you try to do them while you are printing a file. So if you have a long file to print, you might schedule the printing for when you plan to be away from your computer.

The FORMAT Command

When you purchase new diskettes they are blank and unformatted, so you must format them before MS–DOS can use them. Formatting structures a diskette so that MS–DOS can find and store information on it. It also checks the diskette for defective spots. You can format a diskette by using the FORMAT command.

To format a blank diskette in drive B, you would use the following command:

```
FORMAT B: /V
```

You can also format a blank diskette so that some special MS–DOS files are copied onto it during formatting. (If you followed the System Setup instructions for making a simplified MS–DOS System diskette, you have already formatted a diskette and copied these special files to it.) These files are necessary if you want to use the diskette to start MS–DOS. So to format a blank diskette in drive B and include the special MS–DOS files, you would use the following command:

```
FORMAT B: /V /S
```

If you don't want to use the diskette to start MS–DOS, you don't need to specify the /S option. But if you have a diskette and don't know whether you can use it to start MS–DOS, just put the diskette into drive A and press **Ctrl–Alt–Del**. If the diskette doesn't contain the system files, MS–DOS displays an error message.

Suppose you need to create a new data diskette to hold some tax records, but you don't want to copy the special MS–DOS files when formatting the diskette. So to format and label a blank diskette (in drive B) without including the special MS–DOS files, you simply follow these steps:

1. Make sure that the MS–DOS diskette is in drive A.
2. At the MS–DOS prompt, type the following command:

   ```
   FORMAT B: /V
   ```

3. Press the **Enter** key. Your screen should look like this:

   ```
   A>FORMAT B: /V
   Insert new diskette for drive B:
   and strike ENTER when ready
   ```

4. Insert a blank diskette in drive B.
5. Press the **Enter** key to start the format process. When formatting is complete, MS–DOS displays the following prompt:

   ```
   Volume label (11 characters, ENTER for none)?
   ```

6. Type a label that identifies the contents of this diskette (for example, DATA DISK), and press **Enter**. MS–DOS then asks:

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

7. Press **N** (for No) to exit the format program.

Now your diskette is formatted and ready to use. Be sure to label it on the outside cover, and remember to include the volume label that you used in step 6. The label will remind you that you have formatted the diskette, and will help you identify its contents.

CAUTION

The FORMAT program destroys any information that is on a diskette. It's a good idea to list the directory of a diskette that contains files before you format it; then you will be sure that you aren't destroying important files.

The DISKCOPY Command

You may often need to make copies of entire diskettes instead of individual files. You can do this easily with the MS–DOS DISKCOPY command. To use the DISKCOPY command, you must have

- a diskette you want to copy
- a blank diskette to put the copy on.

To copy the contents of a diskette in drive A to a diskette in drive B, you would use the following command:

```
DISKCOPY A: B:
```

Suppose you want to bring a data diskette with you on a business trip, but you don't want to take your original diskette because it might get damaged. All you have to do is use the DISKCOPY command to make a copy of the diskette. For example, to copy the contents of a diskette in drive A to a diskette in drive B, you simply follow these steps.

1. Put the diskette you want to copy in drive A.
2. Put a blank diskette in drive B.
3. At the MS–DOS prompt, type the following command:

   ```
   DISKCOPY A: B:
   ```

4. Press the **Enter** key.

   Your screen should look like this:

   ```
   A>DISKCOPY A: B:
   Insert SOURCE diskette in drive A:
   Insert TARGET diskette in drive B:
   Press any key when ready . . .
   ```

5. Press the space bar to start the DISKCOPY process. When the diskette has been copied, MS–DOS asks:

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

6. Press **N** (for No) to exit the DISKCOPY program.

If you'd like more detailed information about the commands described above, see Section 4, "MS–DOS Commands."

HOW TO RUN APPLICATION PROGRAMS

MS–DOS lets you run many different application programs, including spreadsheets, word processing programs, and graphics packages. These application programs can help you in a number of ways. For instance, they can help you balance a budget, figure income taxes, or manage information, such as inventories, stocks, and address lists.

Once you have started MS–DOS, you can run an application program, as follows:

1. Place the application diskette in an empty diskette drive.
2. Change the default drive to the one that contains the application diskette.
3. Type the name of the application program you want to run.

(If you have a hard disk, you may be able to install the application program on it and run the program from the hard disk without using the application diskette. See the installation instructions for the application program.)

For example, if you had a word processor called Phrase, you might want to use it to write a monthly status report. Suppose also that to start Phrase you must enter the word "PHRASE."

So, to create this report, you would follow these steps:

1. Put your Phrase program diskette into drive B.
2. Change your default diskette drive to B by typing the following:

   B:

3. Start Phrase by entering **PHRASE**.

Then you could use Phrase to create, edit, format, or print your status report.

INTERNAL AND EXTERNAL COMMANDS

There are two types of MS–DOS commands: *internal* commands and *external* commands.

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

BREAK
CHCP
CHDIR
CLS
COPY
CTTY
DATE
DEL
DIR
ECHO
EXIT
FASTOPEN
FOR
GOTO
IF
MKDIR
PATH
PAUSE
PROMPT
REM
REN
RMDIR
SET
SHIFT
TIME
TYPE
VER
VERIFY
VOL

Some internal commands can use paths and pathnames (a sequence of directory names followed by a filename––see also Section 3). Specifically, four commands--COPY, DIR, DEL, and TYPE--have greater flexibility when you specify a pathname after the command.

The formats of these commands are as follows:

COPY *pathname pathname*

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

```
COPY \USER\PETE\*.* SALES
```

DEL *pathname*

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

```
DEL \USER\PETE
```

DIR *pathname*

The following command displays the directory for a specific pathname:

```
DIR \USER\PETE
```

TYPE *pathname*

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

```
TYPE \USER\EMILY\REPORT.NOV
```

Any file name with an extension of .COM, .EXE, or .BAT is considered an *external* command. For example, files such as FORMAT.EXE and DISKCOPY.EXE are external commands. And, because these commands are also files, you can create new commands and add them to MS–DOS. Programs that you create with most languages (including assembly language) will be .EXE (executable) files. Note, however, that when you use an external command, you do not need to type its file name extension.

If you have more than one external command with the same name, MS–DOS will run only one of them, according to the following order of precedence: .COM, .EXE, .BAT.

Suppose, for example, that your diskette includes the files FORMAT.EXE and FORMAT.BAT. If you were to type the external command format, MS-DOS would always run the program FORMAT.EXE first. To run the batch file FORMAT.BAT, you would have to place it in a separate directory and give a path along with the external command.

The following external commands are described in Section 4, "MS-DOS Commands":

APPEND
ASSIGN
ATTRIB
BACKUP
CHKDSK
COMMAND
COMP
CRTDUMP
DISKCOMP
DISKCOPY
EXE2BIN
FDISK
FIND
FORMAT
GRAFTABL
GRAPHICS
JOIN
KEYB
LABEL
MODE
MORE
NLSFUNC
PRINT
RECOVER
REPLACE
RESTORE
RETRACT
SELECT
SHARE
SORT
SUBST
SYS
TREE
TURBO
XCOPY

Before MS-DOS can run external commands, it must read them into memory from the disk. When you give an external command, MS-DOS immediately checks your working directory to find that command. If it isn't there, you must tell MS-DOS which directory the external command is in. You do this with the PATH command.

When you are working with more than one directory, you may find it more convenient to put all the MS-DOS external commands in one directory. Then, when it needs them, MS-DOS can quickly find the external commands at one location.

Suppose, for example, that you are in a working directory named \USER\PROG and that the MS-DOS external commands are in \BIN. To find the FORMAT command, you must tell MS-DOS to choose the \BIN path, as in the following command, which tells MS-DOS to search in your working directory and in the \BIN directory for all commands.

```
PATH \BIN
```

You need only specify this path once during each computer session. Also, if you want to know what the current path is, you can simply type the PATH command by itself. In response, MS-DOS displays the working path on the screen.

You can automatically set your path when you start MS-DOS by including the PATH command in a file called AUTOEXEC.BAT. Refer to Section 5, "Batch Processing," for more information on the AUTOEXEC.BAT file.

REDIRECTING COMMAND INPUT AND OUTPUT

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

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

```
DIR
```

The DIR command can send this output to a file named CONTENTS if you type the following:

```
DIR > CONTENTS
```

If the CONTENTS file doesn't exist, MS-DOS creates it and stores your directory listing there. If CONTENTS does exist, MS-DOS replaces what is in the file with the new data.

If you want to append your directory or add one file to another (instead of replacing the entire file), you can use two greater-than signs (>>) to tell MS-DOS to append the output of the command (such as a directory listing) to the end of a specified file. For example, the following command appends your directory listing to an existing file named CONTENTS:

```
DIR >> CONTENTS
```

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

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

```
SORT <NAMES> NAMELIST
```

FILTERS AND PIPES

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

MS–DOS filters include: find, more, and sort. Their functions are as follows:

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

You can redirect the output from a filter into a file, or use it as input for another filter by using pipes. If you want to use the output from one command as the input for another, you can pipe the commands to MS–DOS. Piping is done by separating commands with the pipe symbol, which is a vertical bar (|). The following command, for example, displays an alphabetically sorted listing of your directory on the screen:

```
DIR | SORT
```

The pipe sends all output generated by the DIR command (on the left side of the bar) as input to the SORT command (on the right side of the bar).

You can also use piping with redirection if you want to send the output to a file. For example, the following command creates a file named DIRECT.LST on your default drive:

```
DIR | SORT > DIRECT.LST
```

The DIRECT.LST file now contains a sorted listing of the directory on the default drive.

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

```
DIR | SORT > B:DIRECT.LST
```

A *pipeline* may consist of more than two commands. The following command, for instance, sorts your directory, shows it to you one screenful at a time, and puts "--More--" at the bottom of your screen when there is more output to be seen:

```
DIR | SORT | MORE
```

Since commands and filters can be piped together in many different ways, you will find many uses for them.

Section 3

More About Files and Directories

In this chapter you will learn about:

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

Before you read this chapter, you should already know how to start MS-DOS, format and make backup copies of diskettes, copy and delete files, and run programs. If you are unfamiliar with any of these actions, refer to Section 2.

PROTECTING YOUR FILES

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

You can take simple but effective measures like putting your diskettes away when you're not using them, or write-protecting program diskettes. Also, if your diskettes contain valuable information, you should make backup copies of them on a regular basis. Another way to protect your programs is by installing your equipment in a secure office or work area.

HOW MS-DOS KEEPS TRACK OF YOUR FILES

As you learned in Section 2, MS-DOS stores files in directories. In addition to directories, it uses an area on a disk (diskette or hard disk) called the File Allocation Table. When you format a disk with the FORMAT command, MS-DOS copies this table onto the disk and creates an empty directory, called the *root directory*. So, on each of your disks, the directories store the files, and the File Allocation Table keeps track of their locations. The table also allocates the free space on your disks so that you have enough room to create new files. See Figure 3–1.

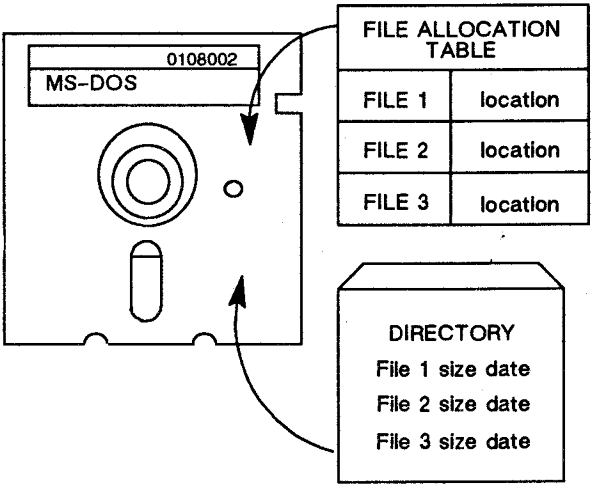


Figure 3–1 File Allocation Table and Directories

These two system areas, the directories and the File Allocation Table, enable MS–DOS to recognize and organize the files on your disks. To check these areas on a disk for consistency and errors you should use the MS–DOS CHKDSK command. When you run CHKDSK, MS–DOS displays a status report and any errors it has found, such as files that show a nonzero size in the directory but that really have no data in them.

For an example of such a display and for more information on CHKDSK, see the description of the CHKDSK command in Section 4, "MS–DOS Commands."

MULTILEVEL DIRECTORIES

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

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

The maximum number of files or directories that an MS–DOS root directory may contain varies, depending on the type of disk drive you are using. Usually, the maximum number is 112 for a double–sided, double–density, 5 1/4–inch floppy disk. The maximum number of entries in the root directory of a 1.44 megabyte, 3 1/2–inch floppy disk is 224. This maximum capacity for a root directory may vary depending upon how the disk is formatted.

These directories may also contain other directories referred to as subdirectories. The number of subdirectories on a disk is not restricted.

This organized file structure is called a multilevel or hierarchical directory system. See Figure 3–2.

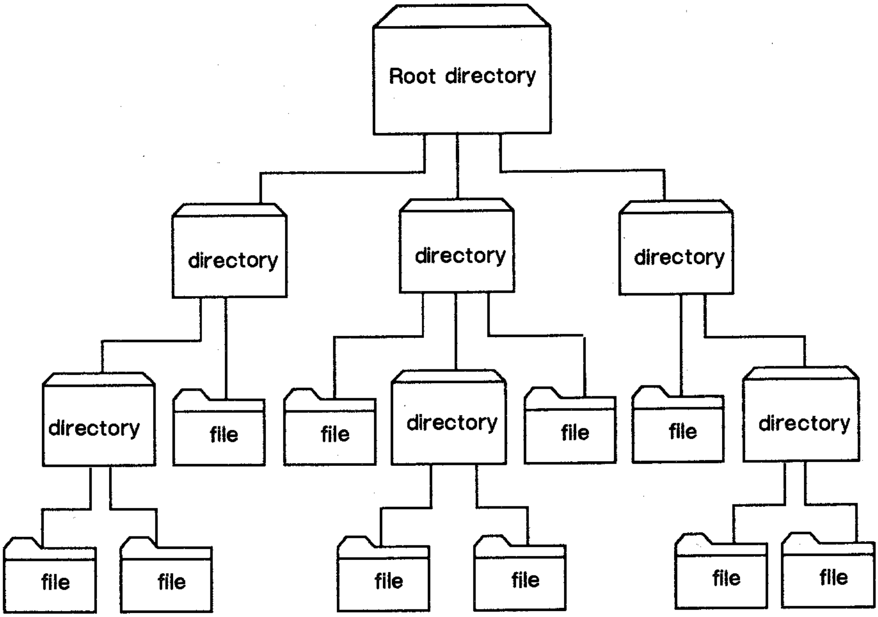


Figure 3–2 Multilevel Directory Structure

The Root Directory

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

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

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

Your Working Directory

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

Because you can put files in different directories, you and your co–workers can have files with the same names, but with unrelated content. Figure 3–3 illustrates a typical multilevel directory structure:

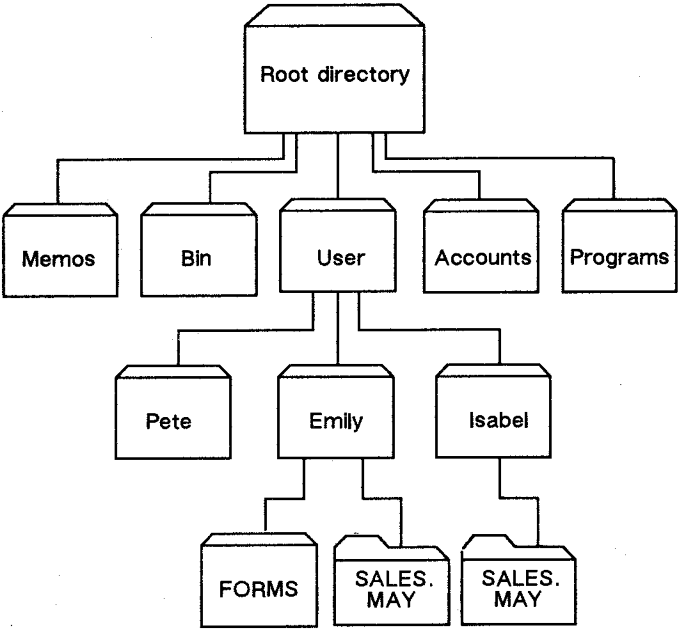


Figure 3–3 Sample Multilevel Directory

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

- A directory of external commands, named BIN.
- A USER directory containing separate subdirectories for all users of the system.
- A directory containing accounting information, named ACCOUNTS.
- A directory of programs, named PROGRAMS.
- A directory of text files, named MEMOS.

As you can see, Pete, Emily, and Isabel each have their own directories, which are subdirectories of the USER directory. Emily has a subdirectory named FORMS, and both Emily and Isabel have SALES.MAY files in their directories, even though Isabel's SALES.MAY file is unrelated to Emily's.

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

```
DIR \USER\EMILY\FORMS
```

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

```
DIR \
```

To find out what files Isabel has in her directory, you would type the following command:

```
DIR \USER\ISABEL
```

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

PATHS

When you use multilevel directories, you must tell MS-DOS where the files are located in the directory system. Both Isabel and Emily, for example, have files named SALES.MAY, so each would have to tell MS-DOS in which directory her file resides when she wants to use it. This is done by giving MS-DOS a *pathname* to the file. A pathname is a sequence of directory names followed by a filename. Each directory name is separated from the previous one by a backslash (\).

The general format of a pathname is as follows:

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

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

The pathname of Emily's SALES.MAY file is:

```
\USER\EMILY\SALES.MAY
```

The pathname of Isabel's SALES.MAY file is:

```
\USER\ISABEL\SALES.MAY
```

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

\ The root directory.

\PROGRAMS
A directory under the root directory that contains program files.

\USER\ISABEL\FORMS\1040
A typical full pathname. This one is for a file named 1040 in the forms directory, which belongs to Isabel.

SALES.MAY
A file in the working directory.

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

MS–DOS uses the shorthand name "." to indicate the name of the working directory in all multilevel directory listings. Two dots ".." are the shorthand name for the working directory's parent directory (one level up). If you enter the DIR command followed by two dots, MS–DOS lists the files in the parent directory of your working directory.

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

```
DIR ..\..
```

WILDCARDS

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

The ? Wildcard

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

```
DIR MEMO?AUG.TXT
```

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

```
MEMO2AUG.TXT
MEMO9AUG.TXT
MEMOBAUG.TXT
```

The * Wildcard

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

```
DIR MEMO*.TXT
```

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

```
MEMO2AUG.TXT
MEMO9AUG.TXT
MEMOBAUG.TXT
MEMOJULY.TXT
MEMOJUNE.TXT
```

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

Aside from the wildcard abbreviation *.*, you should not use more than one asterisk wildcard in a command line. For example, if you type the following command, all the files in the directory will be listed, not just those that contain the number "1":

```
DIR *1*
```

Examples

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

```
DIR A:ACCOUNTS.*
```

Similarly, to list the directory entries for all files with .TXT extensions or in a directory called reports (regardless of their filenames) on the disk in drive B, type the following command:

```
DIR B:\REPORTS\*.TXT
```

This command is useful if your text files have .TXT extensions. For example, by using the DIR command with wildcard characters, you could get a listing of all your text files--even if you don't remember their filenames. For more information on the DIR command, refer to Section 4, "MS-DOS Commands."

USING DIRECTORIES

The following sections describe how to display, change, and delete any directory. You will also learn how to create directories and subdirectories.

How to Create a Directory

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

```
MKDIR USER
```

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

To put files in the new directory, you can use the MS-DOS line editor, EDLIN. Section 7, "The Line Editor (EDLIN)," describes how to use EDLIN to create and save files. You can also create and save files if you have a word processing program such as Microsoft Word.

How to Change Your Working Directory

With MS–DOS it is easy to change from your working directory to a different directory: you simply type the CHDIR (change directory) command followed by a path. For example, if you type CHDIR \USER and then press the **Enter** key, MS–DOS changes the working directory to \USER. You can also specify any path after the command so that you can "travel" around the directory structure. The following command, for example, puts you in the parent directory of your working directory:

```
CHDIR ..
```

How to Display Your Working Directory

All commands are executed while you are in your working directory. You can find out the name of the directory you are in by typing the MS–DOS CHDIR command with no path. For example, if your working directory is \USER\PETE, when you type CHDIR and press the Enter key, you would see the following:

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

This is your working drive, A, plus the working directory, \USER\PETE.

You can also type the letters CD for the CHDIR command to save time. For example, the following commands are the same:

```
CD \USER\PETE
CHDIR \USER\PETE
```

If you want to see the contents of the \USER\PETE directory, you can use the MS–DOS DIR command. The subdirectory might look like this:

```
Volume in drive A has no ID
Directory of A:\USER\PETE

.          <Dir>  08-09-86 10:09a
..         <Dir>  08-09-86 10:09a
TEXT       <Dir>  08-09-86 10:09a
FILE1      TXT 5243 08-04-86 9:30a
4 File(s) 836320 bytes free
```

Note that MS–DOS lists both files and directories in this output. As you can see from the display, Pete has a subdirectory named TEXT; the "." refers to the working directory \USER\PETE; the ".." is short for the parent directory \USER, and FILE1.TXT is a file in the \USER\PETE directory. All these directories and files are on the diskette in drive A.

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

How to Delete a Directory

If you create a directory and decide later that you don't want it any more, you can delete it with the MS-DOS RMDIR (remove directory) command.

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

To remove all the files in a directory (except for the "." and ".." entries), type DEL followed by the path of the directory. For example, to delete all files in the \USER\EMILY directory, type the following command:

```
DEL \USER\EMILY
```

MS-DOS prompts you with the following message:

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

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

Now you can use the RMDIR command to delete the \USER\EMILY directory by typing the following command:

```
RMDIR \USER\EMILY
```

To save time you can also use the letters RD for the RMDIR command.

How to Rename a Directory

There is no command to rename a directory in MS-DOS. You can, however, rename a directory that has no subdirectories. Suppose, for example, you want to rename the \USER\PETE directory and call it \USER\EMILY. To do this you would follow these steps (remember to press the **Enter** key after each step):

1. To create the new directory, type:

   ```
   MKDIR \USER\EMILY
   ```

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

   ```
   COPY \USER\PETE\*.* \USER\EMILY
   ```

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

   ```
   DEL \USER\PETE\*.*
   ```

   (Type **Y** in response to the prompt "Are you sure?")

4. Finally, to remove the old directory, type:

   ```
   RMDIR \USER\PETE
   ```

Section 4

MS-DOS Commands

ABOUT THIS SECTION

This section lists the MS-DOS commands in alphabetical order. For each command the syntax is given along with detailed descriptions, comments, and examples.

External and internal commands are indicated by the following symbols:

I	This shows that the command is internal.
E	This shows that the command is external.

Some of the MS-DOS commands do not work over a computer network. If you try to use these commands, MS-DOS displays the error message "Cannot *command* to a network device," where *command* is the name of the command you typed.

If the command does not work over a network (on a shared or remote device), you will see this symbol in the command description:

NET

The commands that do not work over a network are:

- CHKDSK
- DISKCOMP
- DISKCOPY
- FDISK
- FORMAT
- LABEL
- RECOVER
- SUBST
- SYS

COMMAND OPTIONS

Command options give MS–DOS extra information about a command. If you do not include some options, MS–DOS provides values called *default values*. For the default values of particular commands, you should refer tc individual command descriptions in this section.

MS–DOS commands use the following syntax:

command [*options*]

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

drive:

Refers to a disk drive name. You need only specify a drive name if you are using a file that is *not* on the default drive.

filename

Refers to any file and includes the filename extension (if there is one). The filename option does *not* refer to a device or drive name.

pathname

Refers to a filename by the following syntax:

[\directory] [\directory...]\filename

path

Refers to a directory name by the following syntax:

[\directory] [\directory...]\directory

switches

Control MS–DOS commands. Switches begin with a slash; for example, /P.

arguments

Provide more information to MS–DOS commands. You usually choose between arguments; for example: ON or OFF.

MORE ABOUT OPTIONS

This guide uses the following conventions for command options:

- You must supply the text for any of the variable items shown in italics. For example, when *filename* is shown, you should type the name of your file.
- Items in brackets ([ ]) are optional. If you want to include optional information, you should only type the information within the brackets. Do not type the brackets.
- An ellipsis (...) means that you can repeat an item as many times as necessary.
- You must separate commands from their options by inserting certain characters or spaces. These characters are sometimes called separators. Generally, you should use spaces to separate commands from their options; for example:

  ```
  RENAME DULL.DOC SHARPE.DOC
  ```

- You can also use a semicolon (;), an equal sign (=), or a tab to separate MS–DOS commands from their options.

  In this manual, spaces separate commands from their options.

- You should use punctuation marks, such as backslashes, slashes, equal signs, quotation marks, or colons, where appropriate.
- Disk drives are referred to as *source* drives and *target* drives. A source drive is the drive from which you transfer information. A target drive is the drive to which you transfer information.
- Many commands manipulate *strings* of text. A string is a group of characters that can include letters, numbers, spaces, and any other characters. Searching for a particular word in a file is a common use of a string.

MS–DOS COMMANDS

The MS–DOS commands listed in Table 4–1 are described in this section. Note that synonyms for commands are in parentheses. These commands are described in detail in the section.

Table 4-1 MS-DOS Commands

COMMAND	WHAT IT DOES
APPEND	Sets a search path for data files.
ASSIGN	Assigns a drive letter to a different drive.
ATTRIB	Sets or displays file attributes.
BACKUP	Backs up one or more files from one disk to another.
BREAK	Sets Ctrl-C check.
CHCP	Displays or changes the current code page for the command processor COMMAND.COM.
CHDIR	Changes directories or prints the working directory (CD).
CHKDSK	Scans the directory of the default or designated drive and checks for consistency.
CLS	Clears the screen.
COMMAND	Processes internal MS-DOS commands.
COMP	Compares the contents on two files.
COPY	Copies the specified file(s).
CRTDUMP	Prints screen graphics on an NEC Pinwriter.
CTTY	Changes the device from which you enter commands.
DATE	Displays and sets the date.
DEL	Deletes the specified file(s) (ERASE).
DIR	Lists the requested directory entries.
DISKCOMP	Compares disks.
DISKCOPY	Copies disks.
EXE2BIN	Converts executable (.EXE) files to binary format.
EXIT	Exits the command processor and returns to the previous level.
FASTOPEN	Decreases the amount of time needed to open frequently used files and directories.
FC	Compares two files or two sets of files and displays the differences between them.
FDISK	Configures hard disks for MS-DOS.
FIND	Searches for a constant string of text.
FORMAT	Formats a disk to receive MS-DOS files.
GRAFTABL	Loads a table of graphics characters.
GRAPHICS	Prepares MS-DOS for printing graphics.
JOIN	Joins a disk drive to a phathname.
KEYB	Loads a keyboard program.
LABEL	Labels disks.
MKDIR	Makes a directory (MD).
MODE	Sets operation modes for devices.
MORE	Displays output one screen at a time.
NLSFUNC	Loads country-specific information.
PATH	Sets a command search path.
PRINT	Prints files.
PROMPT	Allows you to change the prompt.

Table 4–1 MS–DOS Commands (cont'd)

COMMAND	WHAT IT DOES
RECOVER	Recovers a bad disk or file.
REN	Renames first file as second file (RENAME).
REPLACE	Replaces previous versions of files.
RESTORE	Restores backed up files.
RETRACT	Moves all hard disk heads to a safe landing zone.
RMDIR	Remove a directory (RD).
SELECT	Selects keyboard layout, date, and time format.
SET	Sets one string value to another in the environment, or displays the environment.
SHARE	Installs file sharing and locking.
SORT	Sorts data forward or backward.
SUBST	Substitutes a string for a pathname.
SYS	Transfers MS–DOS system files from one drive to the the drive specified.
TIME	Displays and sets the time.
TREE	Displays directory and file names.
TURBO	Sets the processor speed to high or low.
TYPE	Displays the contents of a file.
VER	Prints the MS–DOS version number.
VERIFY	Verifies all writes to a disk.
VOL	Displays the volume label.
XCOPY	Copies files and subdirectories.

APPEND

E

Purpose

Sets a search path for data files.

Syntax

APPEND [/X] [/E]

APPEND [*drive*:]*path*[;[*drive*:][*path*]...]

APPEND ;

Comments

The APPEND command allows you to specify a search path for data files. The APPEND command accepts switches only the first time the command is invoked.

Switches

/X Extends the search path for data switches files. MS-DOS first searches the current directory for data files. If MS-DOS doesn't find the needed data files there, it searches the first directory in the APPEND search path. If the files are still not found, MS-DOS continues to the second appended directory, and so on. MS-DOS will not search subsequent directories once the data files are located.

/E Causes appended directories to be stored in the MS-DOS environment.

You can specify more than one path to search by separating each with a semicolon (;). If you type the APPEND command with the path option a second time, MS-DOS discards the old search path and uses the new one.

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

To delete appended paths, use the APPEND ; command. MS-DOS sets the NUL data path, which means that MS-DOS searches only the working directory for data files.

Notes

You can use the APPEND command across a network to locate remote data files.

If you are using the MS-DOS ASSIGN command, you must use the APPEND command before assign.

If you want to set a search path for external commands, see the path command in this chapter.

APPEND searches the data path for all files, regardless of their file extensions, only with the following MS-DOS system calls:

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

Examples

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

```
APPEND B:\LETTERS;A:\REPORTS
```

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

```
APPEND /X
```

If you then typed the following command, MS-DOS would first search your current directory for data files. If MS-DOS did not find the data files in your current directory, it would search the directory called \NEWORDER on drive C. If the files were not there, MS-DOS would search \BAKORDER on drive C.

```
APPEND C:\NEWORDER;C:\BAKORDER
```

ASSIGN

E

Purpose

The ASSIGN command tells MS–DOS to direct all requests from one drive to another.

Syntax

ASSIGN [*x*=*y*] [...]

Comments

Enter only the drive letters. Do not enter the colons on the command line.

x is the source drive, *y* is the new destination drive.

Entering the ASSIGN command without any original or destination parameters resets the current assignments to the default values.

Use ASSIGN only when necessary to access a different drive.

ASSIGN can be used in an AUTOEXEC.BAT file.

The following restrictions apply to ASSIGN.

- Do not use ASSIGN with the following utilities that require actual device information: BACKUP, JOIN, LABEL, PRINT, RESTORE, and SUBST.
- Do not use ASSIGN with DISKCOPY and DISKCOMP because these utilities ignore drive substitutions.

Examples

The command

```
ASSIGN A=C
```

routes all requests for drive A to drive C.

```
ASSIGN A=C B=C
```

routes all requests for drive A and drive B to drive C.

If you type

```
ASSIGN
```

and press **Enter**, you delete all drive assignments. All requests for drive A will go to drive A.

E

ATTRIB

Purpose

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

Syntax

ATTRIB [[+1-]R][[+1-]A] [*drive*:]*pathname* [/S]

Options

+R sets the read-only attribute of a file.

-R disables read-only mode.

+A sets the archive attribute of a file.

-A clears the archive attribute of a file.

Comments

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

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

The BACKUP, RESTORE, and XCOPY commands use the archive attribute as a control mechanism. You can use the +A and -A options to select files that you want to back up with the BACKUP /M command, or copy with the XCOPY /M or XCOPY /A commands.

Notes

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

Examples

To display the attribute of a file called NEWS86 on the default drive, you would type the following command:

```
ATTRIB NEWS86
```

The following command gives the file REPORT.TXT read-only permission:

```
ATTRIB +r REPORT.TXT
```

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

To remove read-only permission from the files in the \USR\PETE directory on drive B, and the files in any subdirectories, you would type the following command:

```
ATTRIB -R B:\USR\PETE
```

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

```
ATTRIB +A A:*.*
```

```
ATTRIB -A A:*.BAK
```

and:

```
XCOPY A: B: /M
```

or

```
COPY A: B: /A
```

If you use the XCOPY /M switch, XCOPY automatically turns off the archive bits of the files in drive A as it copies them.

E

BACKUP

Purpose

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

Syntax

BACKUP [*drive1*:] [*path*] [*filename*] [*drive2*:]
[/S] [/M] [/A] [/F] [/D:date] [/T:time]
[/L:[[*drive*:] [*path*]*filename*]]

Options

drive1 is the disk drive that you want to backup.

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

Comments

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

Switches

Switch	Purpose
/S	Backs up subdirectories.
/M	Backs up only those files that have changed since the last backup.
/A	Adds the files to be backed up to those already on the backup disk. It does not erase old files on the backup disk. This switch will not be accepted if files exist that were backed up using BACKUP from MS–DOS version 3.2 or earlier.
/F	Causes the target disk to be formatted if it is not already. For this switch to function, the MS–DOS FORMAT command must be accessible by the current path.
/D:date	Backs up only those files that you last modified on or after date.
/T:time	Backs up only those files that you last modified at or after time.

/L:filename	Makes a backup log entry in the specified file. If you do not specify filename, BACKUP places a file called BACKUP.LOG in the root directory of the disk that contains the files being backed up.

A backup log file uses the following format:

The first line lists the date and time of the backup.

A line for each backed–up file lists the filename and number of the backup disk on which the file resides.

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

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

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

Notes

You cannot use an old version of the RESTORE command (MS–DOS 3.2 or earlier) for files backed up with the MS–DOS 3.3 BACKUP command.

Unless you use the /A switch, BACKUP erases the old files on a backup disk before adding new files to it.

You should not use the BACKUP command if the drive you are backing up has been assigned, joined, or substituted with the ASSIGN, JOIN, or SUBST commands. If you do, you may not be able to restore the files with the RESTORE command.

The BACKUP program returns the following exit codes:

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

You can use the batch processing IF command for error processing that is based on the error level returned by BACKUP.

Examples

Suppose Emily wants to back up all the files in the \USER\EMILY directory on drive C to a blank, formatted disk in drive A. To do this, she would type

```
    BACKUP C:\USER\EMILY A:
```

BREAK

Purpose

Sets Ctrl–C check.

Syntax

BREAK [ON]

or

BREAK [OFF]

Comments

Depending on the program you are running, you may use Ctrl–C to stop an activity (for example, to stop sorting a file). Normally, MS–DOS checks to see whether you press Ctrl–C while it is reading from the keyboard or writing to the screen or printer. If you set break to ON, you extend Ctrl–C checking to other functions such as disk reads and writes.

Examples

To check for Ctrl–C only during screen, keyboard, and printer reads and writes, type the following:

```
BREAK OFF
```

To find out how break is currently set, type the BREAK command and press **Enter**.

Some programs may set themselves to respond to Ctrl–C at any time. Setting break does not affect these programs.

CHCP

I

Purpose

Displays or changes the current code page for the command processor COMMAND.COM.

Syntax

CHCP [*nnn*]

Comments

The nnn value is the code page to start.

The CHCP command accepts one of the two prepared system code pages as a valid code page. An error message is displayed if a code page is selected that has not been prepared for the system.

If you type the CHCP command without a code page, CHCP displays the active code page and the prepared code pages for the system.

You may select any one of the prepared system code pages defined by the COUNTRY command in CONFIG.SYS. The following are valid code pages:

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

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

To see what the current code page setting is, you simply type

```
CHCP
```

CHCP Messages

MS-DOS will respond with a message similar to the following:

```
Active code page: 850
Prepared system code pages: 850 437
```

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

```
Code page 850 not prepared for system
Active code page: 437
Prepared system code pages: 437 865
```

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

```
Code page 850 not prepared for device xxx
```

Notes

In the preceding example, the CHCP command will still change the active code page even if the selected code page is not prepared for a device. If you want to change the active code page to the original code page, you must reissue the CHCP command with the original code page selected.

Example

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

```
CHCP 863
```

CHDIR (CHANGE DIRECTORY)

I

Synonym

CD

Purpose

Changes a directory to a different path; displays the working directory.

Syntax

CHDIR [*path*]

Comments

If your working directory is \USER\EMILY and you want to change your path to another directory (such as \USER\PETE), type the following command and press **Enter**.

```
CHDIR \USER\PETE
```

MS–DOS puts you in the new directory. There is also a shorthand notation for the CHDIR command:

```
CD ..
```

This command always puts you in the parent directory of your working directory.

Examples

If you use CHDIR without a path, you can display the name of your working directory. For example, if your working directory is \USER\PETE on drive B, and you type the CHDIR command, then press **Enter**, MS–DOS displays the following:

```
B:\USER\PETE
```

The following command displays the name of the working directory on drive B:

```
CHDIR B:
```

CHKDSK (CHECK DISK)

E NET

Purpose

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

Syntax

CHKDSK [*drive*:] [*pathname*] [/F] [/V]

Comments

You should run CHKDSK occasionally on each disk to check for errors. If you do run CHKDSK on a disk and any errors are found, CHKDSK displays the error messages, followed by a status report.

A typical status report might look like this:

```
160256 bytes total disk space
8192 bytes in 2 hidden files
512 bytes in 2 directories
30720 bytes in 8 user files
121344 bytes available on disk

65536 bytes total memory
53152 bytes free
```

The /F switch fixes errors on the disk. If you do not specify this switch, CHKDSK does not correct errors that it finds in your directory, though it does display messages about files that need to be fixed.

If you specify the /V switch, CHKDSK displays messages while it is running.

If you type a filename after CHKDSK, MS–DOS displays a status report for the disk and for the individual file.

CHKDSK does not correct errors on a disk unless you specify the /F switch. For more information on CHKDSK errors, refer to the specific error message in Appendix F, "MS–DOS Message Directory."

Examples

If you want to save the CHKDSK status report for future use, you can redirect the output from CHKDSK to a file by typing the following:

```
CHKDSK A:>FILENAME
```

The errors are then sent to the specified file. Remember, though, not to use the /F switch when you redirect chkdsk output.

If CHKDSK finds errors on the disk in drive A and you want to try to correct them, type the following command:

```
    CHKDSK A: /F
```

CHKDSK now tries to correct any errors it encounters on the disk in drive A.

CLS

Purpose

Clears the terminal screen.

Syntax

CLS

Comment

This command clears your terminal screen by sending the ANSI escape sequence (ESC[2J) to your console.

Example

To clear your screen, type the following and press **Enter**:

```
CLS
```

COMMAND

E

Purpose

Starts the command processor.

Syntax

COMMAND[*drive*:][*path*][*cttydev*][/E:*nnnnn*][/*P*][/C *string*]

Comments

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

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

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

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

cttydev allows you to specify a different device (such as AUX) for input and output. See the CTTY command in this section for more information.

The /E switch specifies the environment size, where *nnnnn* is the size in bytes. The size may range between 128 and 32768 bytes. The default value is 128 bytes.

If *nnnnn* is less than 128 bytes, MS–DOS defaults to 128 bytes and displays the following message:

```
Invalid environment size specified
```

If *nnnnn* is greater than 32768 bytes, MS–DOS displays the same message, but defaults to 32768 bytes.

The /P switch tells COMMAND.COM not to exit to any higher level shell.

The /C switch, if used, should be the last switch in the command. It tells the command processor to perform the command or commands specified by *string* and to then return.

Example

The following command tells the MS–DOS command processor to do three things.

- Start a new command processor under the current program.
- Run the command "CHKDSK B:"
- Return to the first command processor.

```
COMMAND /C CHKDSK B:
```

To learn how to use a pathname and the /P switch with command, see the sample CONFIG.SYS file in Appendix C, "How to Configure Your System."

COMP

E

Purpose

The COMP command compares the contents of a file or group of files with another file or group of files to verify that they are identical.

Syntax

COMP[*drive*:] [*path*] [*filename*[.*ext*]] [*drive*:] [*path*] [*filename*[.*ext*]]

Comments

COMP compares two sets of files; DISKCOMP compares the contents of two diskettes.

The first pathname, [*drive*:] [*path*] [*filename*[.*ext*]], is the file or group of files that is to be compared with the second pathname, [*drive*:] [*path*] [[*filename*[.*ext*]].

If you do not specify a disk drive, COMP uses the default drive.

If you do not specify a filename, COMP uses *.*.

If you specify a disk drive for the second file only, COMP assumes the second filename is the same as the first filename.

Only files that have names matching those specified in the first pathname are compared. You must enter the second pathname for this command to perform as expected.

You can use wildcard characters to specify more than one file to be compared (see "Examples").

Files can be on either the same or different drives. Files can also be in the same or a different directory.

If the files to be compared are identical, the following message appears on your screen:

```
Files compare OK
Compare more files (Y/N)?
```

Enter **N** to stop the COMP command. Enter **Y** to repeat the procedure with another set of files.

If more than one file is being compared, the COMP process continues until all files have been compared.

The COMP command displays an error message if one of these problems occurs:

- a specified directory path is invalid
- two files to be compared are of a different size
- a file specified by the second pathname cannot be found.

If files contain mismatching data, COMP displays the following message:

```
Compare error at offset xxxxxxxx
      File 1 = xx
      File 2 = xx
```

xxxxxx indicates the offset into the files of the mismatching bytes.

If 10 data mismatches are found, the COMP command stops and displays this message:

```
10 mismatches - ending compare
```

Some application programs use files that are recorded in exact 128–byte multiples. Sometimes the actual data in these files may be less than exactly 128 bytes, and the COMP program finds mismatches at the end of these types of files.

If an end–of–file (EOF) marker is not found at the end of a COMP command process, this message appears on your screen:

```
EOF mark not found
```

The EOF message is not an error message; it indicates that COMP may have found random data bytes after valid data, but before the last 128–byte block ends.

Examples

```
COMP A:*.BAS B:*.BAK
```

compares each file with extension .BAS on drive A with a file of the same name, but extension .BAK on drive B.

```
COMP A:*.COM C:
```

compares all .COM files on drive A with the files of the same name and extension on drive C.

```
COMP C:\SUBDIR1 C:\SUBDIR2
```

locates and compares all files in directory SUBDIR1 with the files of the same name in SUBDIR2 on drive C.

COPY

I

Purpose

Copies one or more files to another disk. This command also appends files and copies files on the same disk.

Syntax

COPY [*drive*:]*pathname* [*drive*:][*pathname*][/V][/A][/B] (to copy files)

COPY *pathname* + *pathname* ... (to append files)

Comments

If you do not specify the second *pathname*, the copy is created on the default drive and has the same name as the original file (first *pathname*). If the original file is on the default drive and you do not specify the second *pathname*, the COPY command quits (you are not allowed to copy a file to itself), and MS–DOS displays the following error message:

```
File cannot be copied onto itself
0 File(s) copied
```

If the source and target files are both in the working directory, you may use filenames instead of complete pathnames.

The second *drive:pathname* option may take one of three forms:

- If the second option is a drive name only, MS–DOS copies the original file to the designated drive, keeping the original filename. For example, the following command makes a copy on drive B named MEMO.DOC:

  ```
  COPY MEMO.DOC B:
  ```

- If the second option is a filename only, MS–DOS copies the original file to one on the default drive, and renames it with the specified filename. For example, the following command makes a copy of MEMO.DOC, names it LETTER.DOC, and places it on the default drive:

  ```
  COPY MEMO.DOC LETTER.DOC
  ```

- If the second option includes a drive name, MS-DOS copies the original file to one on the specified drive. For example, the following command makes a copy of MEMO.DOC on the default drive, names the copy LETTER.DOC, and places the copy on the disk in drive B:

```
COPY MEMO.DOC B;LETTER.DOC
```

The /V switch causes MS-DOS to verify that the sectors written on the target disk are recorded properly. If MS-DOS cannot verify a write, it displays an error message. Although there are rarely recording errors when you run COPY, the /V switch lets you verify that critical data has been correctly recorded; it also makes the COPY command run more slowly because MS-DOS must check each entry recorded on the disk.

The /A or /B switch lets you copy either ASCII or binary files, respectively. Each switch applies to the filename preceding it, and to all remaining filenames in the command, until COPY encounters another /A or /B switch.

Examples

When used with a source filename:

/A Causes the file to be treated as an ASCII (text) file. Data in the file is copied up to but not including the first end-of-file mark (in EDLIN this is Ctrl-Z). The remainder of the file is not copied.

/B Causes the entire file to be copied, including any end-of-file marks.

When used with a target filename:

/A Causes an end-of-file character to be added as the last character of the file; for example:

```
COPY MEMO.DOC /A LETTER.DOC
```

/B Does not add an end-of-file character; for example:

```
COPY BILLING.ASM /B BILLING2.ASM
```

When you are appending files the default switch is always /A.

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

```
COPY INTRO.RPT + BODY.RPT + B:SUM.RPT REPORT
```

This command combines files named INTRO.RPT, BODY.RPT, and SUM.RPT (on drive B), and places them in a file called REPORT on the default drive. If you leave out the target file, MS-DOS combines the files into the first specified file.

You can also combine several files into one by using wildcards; for example:

```
COPY *.TXT COMBIN.DOC
```

This command takes all files with an extension of .TXT and combines them into one file named COMBIN.DOC.

In the following example, each file that matches *.TXT is combined with its corresponding .REF file. The result is a file with the same filename but with the extension .DOC. Thus, FILE1.TXT is combined with FILE1.REF to form FILE1.DOC, XYZ.TXT with XYZ.REF to form XYZ.DOC, and so on:

```
COPY *.TXT + *.REF *.DOC
```

The following copy command combines all files matching *.TXT and all files matching *.REF into one file named COMBIN.DOC:

```
COPY *.TXT + *.REF COMBIN.DOC
```

CAUTION

> Do not try to append files if one of the source filenames has the same extension as the target. For example, if the file ALL.TXT already exists, the following command is an error:
>
> ```
> COPY *.TXT ALL.TXT
> ```

MS-DOS would not detect the error until it tried to append ALL.TXT. But at that point, COPY might have already destroyed the ALL.TXT file.

COPY compares the filename of the input file with the filename of the target. If they are the same, that one input file is skipped, and MS-DOS prints the error message "Content of destination lost before copy." Further joining proceeds normally. For example, the following command appends all *.TXT files (except ALL.TXT) to ALL.TXT:

```
    COPY ALL.TXT + *.TXT
```

This command will not produce an error message. If you want to copy files and subdirectories, you should use the XCOPY command. Refer to the XCOPY command in this section for more information on how to do this.

CRTDUMP

Purpose

The CRTDUMP command enables you to print the contents of the video screen, generated by either the Color Graphics Board (CGB), or the Advanced Graphics Board (AGB), on an NEC Pinwriter P5, P6, P7, or P5XL. To be functional, CRTDUMP must first be loaded into the memory of your computer. Then to activate the screen dump function, press and hold Shift, then press Print Screen. The contents of the screen will then be sent to the Pinwriter.

Syntax

CRTDUMP [*printer type*] [/manual mode] [/R] [/B] [/NO]

Comments

[*printer type*] specifies the type of printer that you are using. There are two types of printers:

1. [MONO] specifies NEC Pinwriters P5, P6, and P7. If no printer type is specified, mono is the default.
2. [COLOR] specifies NEC Pinwriter P5XL with a color ribbon.

/R prints black and white as shown on your screen. If you do not specify the /R switch, CRTDUMP prints black as white and white as black. There is no change in the printing of other colors.

/B prints the background color on the Color Pinwriters if the screen is in 320 x 200 four color graphics mode (mode 4 or 5). If you do not specify the /B switch, CRTDUMP does not print the background color.

/NO re-enables any print screen routine which was in place when CRTDUMP was first run.

Example

```
CRTDUMP color /B
```

This command loads CRTDUMP for a color printer and will allow printing of background colors of 320 x 200 four color graphic screens.

```
CRTDUMP /R
```

This command loads CRTDUMP for monochrome printers and prints black and white as they appear on the screen.

Notes

Printing can take several minutes for graphic mode screens and longer on color printers. If the screen is in text mode, CRTDUMP prints the text much more quickly.

Graphic mode screen dumps are printed sideways, i.e. the upper left corner of the screen is printed on the upper right corner of the paper.

The type of output to the printer depends upon both the printer type and screen mode. CRTDUMP output is as follows.

	SCREEN	DISPLAY	PRINTER	
MODE	DESCRIPTION	ADAPTER	MONO	COLOR
0	40 X 25 TEXT (BW)	CGB, AGB	BLACK TEXT	COLOR TEXT
1	40 X 25 TEXT (COLOR)	CGB, AGB	BLACK TEXT	COLOR TEXT
2	80 X 25 TEXT (BW)	CGB, AGB	BLACK TEXT	COLOR TEXT
3	80 X 25 TEXT (COLOR)	CGB, AGB	BLACK TEXT	COLOR TEXT
4	320 X 200 (4 COLOR)	CGB, AGB	GRAY SHADES	4 COLORS
5	320 X 200 (4 COLOR BW)	CGB, AGB	GRAY SHADES	4 COLORS
6	640 X 200 (MONO)	CGB, AGB	BLACK/WHITE	BLACK/WHITE
7	80 X 25 TEXT (MONO)	CGB, AGB	BLACK TEXT	BLACK TEXT
13	320 X 200 (16 COLOR)	AGB	GRAY SHADES	16 COLORS
14	640 X 200 (16 COLOR)	AGB	GRAY SHADES	16 COLORS
15	640 X 350 (MONO)	AGB	BLK/GRY/WHT	BLK/GRY/WHT
16	640 X 350 (16 COLOR)	AGB	GRAY SHADES	16 COLORS

If the screen is in 320 x 200, 640 x 200 or 640 x 350 sixteen color graphics mode, the screen is printed in up to 16 colors (color Printers). The 16 colors consist of 8 colors + 8 intensified colors. The 8 base colors are black, blue, green, cyan, red, magenta, yellow and white.

CRTDUMP is a resident program which decreases the amount of free memory in the system. Running CRTDUMP more than once, however, to change the parameters, will not cause additional copies of the resident program to be loaded.

CTTY

Purpose

Lets you change the device from which you issue commands.

Syntax

CTTY *device*

Comments

The device parameter specifies the device from which you are giving commands to MS-DOS. CTTY is useful if you want to change the device on which you are working. In this command, the letters TTY represent the console; that is, your keyboard.

Examples

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

```
CTTY AUX
```

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

```
CTTY CON
```

There are many programs that do not use MS-DOS for input, output, or either. These programs send input directly to the hardware on your computer. The CTTY command has no effect on these programs; it affects only programs that use MS-DOS.

DATE

I

Purpose

Enters or changes the date known to the system.

Syntax

DATE [*mm-dd-yy*]

Comments

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

Remember to use only numbers when you type the date; allowed numbers are:

mm	=	1-12
dd	=	1-31
yy	=	80-79 or 1980-2079

The date, month, and year entries may be separated by hyphens (-) or slashes (/). MS-DOS is programmed to change months and years correctly, whether the month has 31, 30, or 28 days-- or 29 days, since MS-DOS handles leap years, too.

It is possible for you to change the *mm-dd-yy* format in which the date is displayed and entered. The COUNTRY command in the CONFIG.SYS file allows you to change the date format to the European standard *dd-mm-yy*. For more information on the CONFIG.SYS file, see Appendix C, "How to Configure Your System."

Notes

This command sets the internal clock in your computer, if one exists.

The format *mm-dd-yy* may vary if you are using a code page other than the one for the United States. For more information about international date formats, see Appendix E, "How to Use Code Pages."

Examples

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

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

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

```
DATE 3–9–86
```

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

DEL (DELETE)

Synonym

ERASE

Purpose

Deletes all files specified by the drive and pathname.

Syntax

DEL [*drive*:]*pathname*

Comments

You can use the * and ? wildcards to delete more than one file at a time.

If the *pathname* is *.* the prompt "Are you sure?" appears. If you then type. Y as a response, all files on the disk are deleted.

CAUTION

> Once you have deleted a file on your disk,
> you cannot recover it.

Examples

The following command deletes a file named VACATION:

```
DEL VACATION
```

If you have two files named VACATION.FEB and VACATION.APR, you can delete them both with the following command:

```
DEL VACATION.*
```

DIR (DIRECTORY)

Purpose

Lists the files in a directory.

Syntax

DIR [*drive*:] [*pathname*] [/P] [/W]

Comments

If you just type the DIR command by itself, all directory entries on the default drive are listed. If you include a drive name in the command, such as B:, all entries in the default directory of the disk in drive B are listed. If you include a filename without an extension (INVOICES, for example), MS–DOS lists all files named INVOICES in the default directory of the disk in the default drive.

When you use the DIR command with a filename and drive letter (B:INVOICES, for example), MS–DOS displays all files on the disk in drive B with the filename INVOICES. In all cases, (except when using the /W switch), MS–DOS lists files with their size in bytes and with the time and date of their last modification.

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

COMMAND	EQUIVALENT
DIR	DIR *.*
DIR *filename*	DIR *filename.**
DIR *.ext*	DIR **.ext*

You may use the /P switch and the /W switch with DIR.

The /P switch used with DIR selects page mode. It also makes the directory display stop scrolling when the screen has filled. To resume scrolling the display, press any key.

The /W switch selects wide display and causes MS–DOS to display only filenames and not other file information. The wide display lists up to five files per line.

Note that if the COUNTRY command in the CONFIG.SYS file is set to a country other than the U.S., the directory date and time formats may differ.

For more information on the CONFIG.SYS file, see Appendix C, "How to Configure Your System."

Example

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

```
    DIR /P
```

This command displays the directory one screenful at a time.

DISKCOMP

E NET

Purpose

The DISKCOMP command compares the contents of the diskette in the first-specified drive to the contents of the diskette in the second-specified drive. Use DISKCOMP after a DISKCOPY operation to ensure that the two diskettes are identical.

Syntax

DISKCOMP [*drive*:][*drive*:][/1][/8]

Comments

DISKCOMP compares diskettes contents only. DISKCOMP does not work with hard disks or with individual files. Use COMP to compare file contents.

The first drive specifies the source diskette drive. The second drive specifies the destination diskette drive. If you do not specify both parameters, DISKCOMP uses the default drive to perform a one-drive comparison. If you do not specify the second parameter, DISKCOMP uses the default drive as a destination drive.

If you use DISKCOMP from a hard disk, you must specify a source and destination drive.

Options

/1 causes DISKCOMP to compare only the first side of diskettes even if the diskettes are double-sided.

/8 causes DISKCOMP to compare only 8 sectors per track even if the diskette contains 9 or 15 sectors per track.

These options are used when diskettes formatted with the /1 or /8 option are compared (see the FORMAT command for details).

Notes

Do not use DISKCOMP to compare diskettes on drives that you have reassigned with ASSIGN or SUBST commands. Such substitutions can hide the true drive letter that DISKCOMP uses.

Do not use DISKCOMP while a JOIN is in effect. JOIN connects pathnames to drive letters, and thereby invalidates the directory structure.

Do not use DISKCOMP with network drives because DISKCOMP is not fully compatible with network software.

DISKCOPY

E ~~NET~~

Purpose

The DISKCOPY command copies all of the files on the source drive to a destination drive. The diskettes must be in separate drives unless your computer has a single diskette drive.

If the diskette in the destination drive is unformatted, DISKCOPY automatically formats the diskette.

Syntax

DISKCOPY [*drive*:] [*drive*:][/1]

Comments

The first drive is the source diskette drive, and the second is the destination drive.

If you omit the second drive, DISKCOPY uses the default drive as the destination drive.

If you specify the same drive for destination and source, DISKCOPY performs a single–drive operation.

If you use DISKCOPY from a diskette and use DISKCOPY with no parameters, the following copy procedures are performed:

- DISKCOPY performs a single–drive copy on the default drive for a dual–diskette drive system.
- DISKCOPY performs a single–drive copy on drive A for a single–diskette drive system.

With a single–drive system, you are prompted to insert diskettes at the appropriate times. DISKCOPY pauses between insertions until you press any key.

If you use DISKCOPY from a hard disk, you must specify both source and target drives.

After copying, DISKCOPY prompts

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

To copy another diskette, enter **Y**. To end DISKCOPY, enter **N**.

Option

The /1 switch copies the first side of the diskette only, even if you have a system with double-sided drives.

Diskette/Drive Agreement

Table 4–2 describes the type of diskette copies that DISKCOPY performs on specified drives.

Table 4–2 DISKCOPY Disk/Drive Agreement

DRIVE CAPACITY	DISKETTE COPIES
Single-sided	Single-sided to single-sided
Double-sided	Single-sided to single-sided Double-sided to double-sided
High capacity (1.2 MB)	Single-sided to single-sided Double-sided to double-sided Double-tracked to double-tracked High capacity to high capacity

CAUTION

Diskettes that have been formatted on or written to a high–capacity (1.2 MB) drive cannot be read on a low–capacity drive (e.g., 360 KB).

Notes

Disks on which you have created and deleted many files can become fragmented because DOS does not allocate disk space sequentially. A fragmented disk performs slowly due to delays from finding, reading, or writing a file. Use the COPY command (instead of DISKCOPY) to copy a diskette that has had a lot of activity.

Do not use DISKCOPY to copy disks to or from drives that you have reassigned with ASSIGN or SUBST commands. Drive substitutions can hide the true drive letter that DISKCOPY uses.

Do not use DISKCOPY while a JOIN is in effect. JOIN connects pathnames to drive letters, and thus can invalidate directory structures.

Do not use DISKCOPY with network drives because DISKCOPY is not fully compatible with network software.

EXE2BIN

Purpose

Converts .EXE (executable) files to binary format.

Syntax

EXE2BIN [*drive*:]*pathname* [*drive*:]*pathname*

Comments

This command converts .EXE (executable) files to binary format. The first *pathname* is the input file; if you do not specify an extension, it defaults to .EXE. The input file is converted to a .BIN file format (a memory image of the program) and placed in the output file (the second *pathname*).

If you do not specify a drive name, EXE2BIN uses the drive of the input file. Similarly, if you do not specify an output filename, EXE2BIN uses the input filename. And finally, if you do not specify a filename extension in the output filename, EXE2BIN gives the new file the extension .BIN.

Some restrictions do apply when you use the EXE2BIN command: the input file must be in valid .EXE format produced by the linker; the resident, or actual code and data part of the file must be less than 64K; and there must be no STACK segment.

With EXE2BIN, two kinds of conversions are possible, depending on whether the initial CS:IP (Code Segment:Instruction Pointer) is specified in the .EXE file.

- If the CS:IP is not specified in the .EXE file, EXE2BIN assumes you want a pure binary conversion. If segment fixups are necessary (that is, if the program contains instructions requiring segment relocation), the command E prompts you for the fixup value. This value is the absolute segment at which the program is to be loaded. The resulting program will be usable only when loaded at the absolute memory address specified by your application. The command processor will not be able to load the program.
- If the CS:IP is 0000:100H, EXE2BIN assumes that the file will 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, since .COM files must be segment relocatable; that is, they must assume the entry conditions explained in the Microsoft Macro Assembler manuals (*User's Guide* and *Reference Manual*). Once the conversion is complete, you may rename the output file with a .COM extension. The command processor will then be able to load and execute the program in the same way as the .COM programs supplied on your MS–DOS diskette.

EXIT

Purpose

Exits the COMMAND.COM program (the command processor) and returns to a previous level, if one exists.

Syntax

EXIT

Comments

If you use the MS–DOS COMMAND program to start a new command processor, you can use the EXIT command to return to the old command processor. Also, while running an application program, you can exit to the MS–DOS command processor, and then return to your program.

Refer to the COMMAND command in this section for more information.

Example

If you start a new command processor by typing the following command, you can then return to the previous command processor by typing the EXIT command.

```
COMMAND C:\
```

FASTOPEN

I

Purpose

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

Syntax

FASTOPEN [*drive*:[=*nnn*][...]]

Comments

The *nnn* value is the number of files per disk.

FASTOPEN tracks the location of files and directories on a disk for fast access. Accessing files in a complex directory structure can be time consuming. If you run applications that use several files (such as a data base application), the time to open and close files noticeably degrades your computer's performance. Every time a file or directory is opened, FASTOPEN-records its name and location. Then, if a file or directory recorded by FASTOPEN is reopened, the access time is greatly reduced.

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

Notes

You can invoke the FASTOPEN command only once. If you want to change the FASTOPEN settings, restart MS–DOS.

FASTOPEN needs approximately 40 bytes of memory for each file or directory location it tracks.

Examples

If you want MS–DOS to track the location of up to 100 files on drive C, you could type the following:

```
FASTOPEN C:=100
```

FDISK

E NET

Purpose

Configures hard disks for MS–DOS.

Syntax

FDISK

Comments

FDISK configures a hard disk for use with MS–DOS. Before you can use your hard disk for the first time, you must use FDISK to configure it. (If the System Setup procedure in Appendix A has been followed, this step has already been done.)

FDISK displays a series of menus to help you partition your hard disk for MS–DOS. With the FDISK command,

- Create multiple MS–DOS partitions
- Create large MS–DOS partitions (greater than 32 MB)
- Create a primary MS–DOS partition
- Create an extended MS–DOS partition
- Change the active partition
- Delete a MS–DOS partition
- Display partition data
- Select the next fixed disk drive for partitioning on a system with multiple fixed disks

Notes

FDISK doesn't work on drives used in the SUBST or JOIN commands.

For more information on how to use FDISK, see Appendix H, "Configuring Your Hard Disk (FDISK)."

FIND

Purpose

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

Syntax

FIND [/V] [/C] [/N] *"string"* [*drive*:][*pathname*]

Comments

FIND is a filter that takes as options a *string* (a group of characters) and a series of filenames. After searching the files given on the command line, FIND displays any lines it has found that contain the specified string. In your command line, this string must be enclosed in double quotation marks.

If you do not specify a pathname, FIND takes input from its standard input (usually the keyboard) and displays any lines that contain the specified string.

The switches for FIND are as follows:

/V Displays all lines not containing the specified string.

/C Prints only the count of lines that contain a match in each of the files.

/N Precedes each line by its relative line number in the file.

You must type double quotation marks around a string that already has double quotation marks.

Examples

The following command displays all lines from a file named PENCIL.AD that contain the string "Automatic Pencil Sharpener":

```
FIND "Automatic Pencil Sharpener" PENCIL.AD
```

The next command causes MS–DOS to display all names of the files on the disk in drive B that do not contain the string "date".

```
DIR B: | FIND /V "date"
```

FORMAT

Purpose

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

Syntax

FORMAT drive: [/1][/4][/7][/8][/N:*xx*][/T:*yy*][/V][/S][/B]

Comments

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

CAUTION

> Formatting destroys any previously existing data on a disk and it ignores drive assignments created with the ASSIGN command.

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

Switches

/1 Formats a single-side of the floppy disk.

/4 Formats a 5 1/4-inch, double-sided disk in a high-capacity disk drive. If you are using a single- or double-sided drive, you may not be able to reliably read disks formatted with this switch.

/7 Formats a 5 1/4-inch, middie-capacity (720KB) diskette in a high-capacity drive.

/8 Formats eight sectors per track.

/B Formats the disk, leaving ample space to copy an operating system, such as MS-DOS 3.3.

/S Copies the operating system files listed in the file FORMATS. TBL from the disk in the default drive to the newly formatted disk. The newly formatted disk must be 1.2 megabytes or greater in size; otherwise, FORMAT rejects the command. If the operating system is not on the default drive, FORMAT prompts you to insert a sysem disk in the default drive (or in drive A if the default drive is nonremovable).

/T:tracks Specifies the number of tracks on the disk. This switch formats 3 1/2-inch floppy disk to the number of tracks specified. For 720-KB disks and 1.44-MB disks, this value is 80 (/T:80).

/N:sectors Specifies the number of sectors per track. This switch formats a 3 1/2-inch disk to the number of sectors specified. For 720-KB disks, this value is 9 (/N:9).

/V Causes format to prompt you for a volume label for the disk you are formatting. A volume label identifies the disk and can be up to 11 characters in length (no tabs allowed). An example of a volume label is programs.

When you format a hard disk, FORMAT prompts you to verify the volume label:

```
Enter current Volume Label for drive x:
```

If your hard disk does not have a volume label, press the **ENTER** key. But note that if your hard disk has never been formatted before, or if it has a bad boot sector, FORMAT will not prompt you for a volume label.

If the volume label that you enter does not match the label on the hard disk, FORMAT displays the following message:

```
Invalid Volume ID Format failure
```

Otherwise it continues:

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

Proceed with FORMAT (Y/N)?_
```

If you want to format your hard disk, type **Y** (for Yes) and press the **ENTER** key. If you don't want to format your hard disk, type **N** (for No) and press the **ENTER** key.

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

Notes

You should not use the FORMAT command with drives used in the ASSIGN, JOIN, or SUBST commands, and you cannot format drives over a network.

For more information about formatting your hard disk, see Appendix H, "Configuring Your Hard Disk (FDISK)."

For more information about disk volume labels, see the DIR, LABEL, and VOLUME commands in this chapter.

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

Disk type	Valid switches
160/180K bytes	/1 /4 /8 /B /V /S
320/360K bytes	/1 /4 /8 /B /V /S
720K bytes (5 1/4″)	/7 /V /S
720K bytes (3 1/2″)	/N /T /V /S
1.2 megabytes	/N /T /V /S
1.44 megabytes	/N /T /V /S
hard disk	/V /S

Do not use the /S and /V switches with the /B switch.

The FORMAT command returns the following exit codes:

Code	Function
0	Successful completion
3	Terminated by user (CONTROL–C)
4	Fatal error (any error other than 0, 3, or 5)
5	N response to hard disk prompt, Proceed with format (Y/N)?

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

You can use the SELECT command instead of FORMAT if you want to format a disk with country–specific information. For more information, see the SELECT command later in this chapter.

Examples

To format a floppy disk in drive A and copy the operating system to it, type the following command:

```
FORMAT A:  /S
```

To format a floppy disk in drive A for use with data, type

```
FORMAT A:  /V
```

GRAFTABL

Purpose

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

Syntax

GRAFTABL [*xxx*]
  or
GRAFTABL /status

Comments

The xxx value is a code page identification number. Valid code pages (xxx) include the following:

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

If you type the GRAFTABL command followed by the /status switch, MS–DOS displays the active character set.

After GRAFTABL loads the character table, it displays the following message:

```
Graphics characters loaded
```

Since you can load the graphics table only once each time you start MS–DOS, you might want to put the GRAFTABL command in your AUTOEXEC.BAT file to save time. If you try to load the same table a second time, GRAFTABL displays the following message:

```
Graphics characters already loaded
```

Notes

The GRAFTABL command increases the size of MS–DOS resident in memory.

For more information about using code pages, see the CHCP command in this chapter.

The GRAFTABL command returns the following exit codes:

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

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

Examples

To load a table of graphics characters into memory, type

```
GRAFTABL
```

GRAPHICS

Purpose

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

Syntax

GRAPHICS [*printer*]
[/B] [/P=port] [/R] [/LCD]

Comments

The printer option is one of the following:

Value	Function
COLOR1	Prints on an IBM Personal Computer Color Printer with black ribbon.
COLOR4	Prints on an IBM Personal Computer Color Printer with RGB (red, green, blue, and black) ribbon.
COLOR8	Prints on an IBM Personal Computer Color Printer with CMY (cyan, magenta, yellow, and black) ribbon.
COMPACT	Prints on an IBM Personal Computer Compact Printer.
GRAPHICS	Prints on an IBM Personal Graphics Printer, or IBM Proprinter.

If you do not specify the printer option, GRAPHICS defaults to the GRAPHICS printer type.

The GRAPHICS command accepts the following switches:

Switch	Purpose
/B	Prints the background in color. This option is valid for COLOR4 and COLOR8 printers.
/P=port	Sets the parallel printer port that GRAPHICS sends its output to when you press the **SHIFT–PRINTSCREEN** key combination. The port may be set to 1, 2, or 3; the default setting is 1.
/R	Prints black and white (as seen on the screen) on the printer. The default is to print black as white and white as black.

To print the screen, press the **SHIFT** and **PRINTSCREEN** keys at the same time. If the computer is in 320200 color graphics mode, and if the printer type is COLOR1 or GRAPHICS, GRAPHICS prints the screen contents with up to four shades of gray. If the computer is in 640200 color graphics mode, GRAPHICS prints the screen contents sideways on the paper.

Note

The GRAPHICS command increases the size of MS-DOS resident in memory.

Example

To print a graphics screen on your printer, type the following command:

```
GRAPHICS
```

Then, when the screen displays the infomation you want to print, press the **SHIFT** and **PRINTSCREEN** keys at the same time.

JOIN

Purpose

Joins a disk drive to a specific path.

Syntax

JOIN [*drive*: *drive:path*]

or

JOIN *drive*: /D

Comments

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

If the path does not exist, MS–DOS tries to make a directory with that path. After you give the JOIN command, the first drive name becomes invalid, and if you try to use it MS–DOS displays the "Invalid drive" error message.

Examples

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

```
JOIN D: C:\SALES
```

But the following one will not.

```
JOIN D: C:\SALES\REGIONAL
```

To reverse JOIN ("unjoin"), use the following format.

```
JOIN drive: /D
```

Here *drive*: represents the source drive, and the /D switch turns off the JOIN command.

If you just type the JOIN command by itself, MS–DOS displays the current drives that are joined.

E

KEYB

Purpose

Loads a keyboard program.

Syntax

KEYB [*xx*[,[*yyy*],[[*drive*:][*path*]*filename*]]]

Options

xx is a two–letter country code.

yyy is the code page which defines the character set.

filename is the name of the keyboard definition file.

Comments

The *xx* option is one of the following two–letter codes:

Codes	Keyboard Type	Command
US	United States	KEYBUS (default)
FR	France	KEYBFR
GR	Germany	KEYBGR
IT	Italy	KEYBIT
SP	Spain	KEYBSP
UK	United Kingdom	KEYBUK
PO	Portugal	KEYBPO
SG	Swiss–German	KEYBSG
SF	Swiss–French	KEYBSF
DF	Denmark	KEYBDK
BE	Belgium	KEYBBE
NL	Netherlands	KEYBNL
NO	Norway	KEyBNO
LA	Latin America	KEYBLA
SV	Sweden	KEYBSV
SU	Finland	KEYBSU

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

```
Current keyboard code: FR Code page: 437
Current CON code page: 437
```

You can switch from the KEYB program to the default (United States) keyboard format at any time by pressing **CONTROL-ALT-F1**. You can then return to the memory-resident keyboard program by pressing **CONTROL-ALT-F2**.

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

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

Note

You can also include the appropriate KEYB command in your AUTOEXEC.BAT file so that you won't have to type it each time you start MS-DOS.

Example

To use a German keyboard, type the following command:

```
    KEYBGR
```

LABEL

E ~~NET~~

Purpose

LABEL creates, changes, or deletes a volume label on a disk.

Syntax

LABEL [*drive*:][*volume label*]

Comments

[*drive:*] specifies the drive that contains the disk you want to label. If you omit [*drive*:], LABEL uses the default drive.

[*volume label*] specifies the label you want to use to identify the disk. Use up to 11 characters for a label name.

Options

LABEL permits you to create, change, and delete labels.

Create a label as follows:

1. Enter **LABEL [*drive*:]**.
2. Type the volume label when you receive the following prompt:

   ```
   Volume in drive X has no label
   Volume label (11 characters, ENTER for none)?
   ```

3. Press **Enter** after you type in the label.

Change an existing label as follows:

1. Enter **LABEL [*drive*:]**.
2. Type in the new label when you receive the following prompt:

   ```
   Volume in drive X is xxxxxxxxxxx
   Volume label (11 characters, ENTER for none)?
   ```

3. Press **Enter**.

The new label replaces the old label.

Delete a label as follows:

1. Enter **LABEL [*drive*:]**.
2. Type **Enter** and press **Y** when you receive the following prompts:

```
Volume in drive X is xxxxxxxxxxx
Volume label (11 characters, ENTER for none)?
```

Delete current volume label (Y/N)?

Example

The following command creates a volume label, NEWSLETTER, for a diskette in drive A:

```
LABEL A:NEWSLETTER
```

Notes

Do not use LABEL to name disks on drives that you have reassigned with the ASSIGN or SUBST commands. Drive substitutions can hide true drive letters that LABEL uses.

Do not use LABEL while a JOIN is in effect. JOIN connects drives to pathnames and thus can invalidate drive names.

Do not use LABEL on network drives. LABEL is not fully compatible with network software.

MKDIR (MAKE DIRECTORY)

I

Synonym

MD

Purpose

Makes a new directory.

Syntax

MKDIR [*drive*:]*path*

Comments

With this command you can create a multilevel directory structure. For instance, when you are in your root directory, you can create subdirectories. Remember, though, that when you create directories with MKDIR, they always appear under your working directory unless you explicitly specify a different path with the MKDIR command.

Examples

The following command creates a subdirectory named \USER in your root directory:

```
MKDIR \USER
```

Now, suppose you want to create a directory named PETE under the \USER directory. To do this you could simply type the following command:

```
MKDIR \USER\PETE
```

MODE

E

Purpose

MODE performs four functions:

- Sets printer options.
- Switches display boards and sets options for Color Graphics Boards.
- Sets options for asynchronous communications adapters.
- Routes printer output to an asynchronous communications adapter.
- Sets or displays code pages for parallel printers or your console screen device.

MODE has a command format for each function.

The following subsections present the format, parameters, options, and examples for each MODE function.

FUNCTION 1: SETTING PRINTER OPTIONS

Syntax

MODE LPTn[:][*chars*][,[*lines*][,P]]

Table 4–5 lists MODE parameters and values you can use to set printer options.

Table 4–3 MODE Parameters

PARAMETER		VALUES
n	— printer number	1,2,3
chars	— characters per line	80 or 132
lines	— lines per inch	6 or 8
p	— continuous retry on time–out errors	

Example

```
MODE LPT1:132,8
```

Sets the mode of operation of printer #1 to 132 characters per line and 8 lines per inch vertical spacing.

Default options for the printer are 80 characters per line and 6 lines per inch.

The default settings are LPT1, 80 characters per line, and 6 lines per inch. You can break out of a time-out loop by pressing the **CONTROL-BREAK** key combination.

Note

To stop the retry loop press **Ctrl-Break**.

To stop MODE from continuously retrying time-out errors, use the MODE command without specifying P.

FUNCTION 2: SWITCHING ADAPTERS AND SETTING THE DISPLAY MODE

Syntax

MODE *display* or MODE *display*[,*shift*[,T]]

Comments

display specifies characters per line and monitor adapter.

shift specifies shift display, right or left. Valid values are L (for left) or R (for right).

T requests a test pattern to align the display.

Table 4-6 lists values for n and the tasks these values cause MODE to perform.

Table 4-4 MODE *n* Values

display VALUE	TASKS
40	Sets the display width to 40 characters per line for Color Graphics Board.
80	Sets the display width to 80 characters per line for Color Grahpics Board.
BW40	Switches the active display adapter to the Color Graphics Board, and Sets the display mode to Black and White (disables color) with 40 characters per line.
BW80	Switches the active display adapter to the Color Graphics Board, and sets the display mode to Black and White (disables color) with 80 characters per line.
CO40	Switches the active display adapter to the Color Graphics Board, enables color, and sets the display width to 40 characters per line.
CO80	Switches the active display adapter to the Color Graphics Board, enables color, and sets the display width to 80 characters per line.
MONO	Switches the active display adapter to the monochrome display board. This adapter has a display width of 80 characters per line.

For readability, you can shift the display one character (for 40 columns) or two characters (for 80 columns) in either direction.

If you specify **T** in the MODE command, a prompt asks you if the screen is aligned properly. Enter Y if the alignment is correct. Entering Y ends MODE processing. Enter any character to repeat the shift.

Example

```
MODE 80,R,T
```

sets the mode of operation to 80 characters per line and shifts the display two character positions to the right.

MODE displays the test pattern to permit you to shift the display again without having to reenter the command.

FUNCTION 3: SETTING OPTIONS FOR ASYNCHRONOUS COMMUNICATIONS ADAPTER

Syntax

MODE COMm[:]*baud*,[*parity*],[*databits*],[*stopbits*],[P]]]]

Comments

This format contains two kinds of parameters: Protocol and P.

1. The protocol parameters are: m, baud, parity, data bits and stop bits. Use these parameters to set the asynchronous communications adapter.

   You must specify a value for the COM port number and the baud rate. You can specify values for parity, data bits, and stop bits, or you can use the default values by simply entering commas.

Table 4-7 lists protocol parameters for this MODE format.

Table 4-5 MODE Protocol Parameters

PROTOCOL PARAMETER	VALUE	DEFAULT
m (adapter number)	1, 2, 3, or 4	COM1
baud rate (bits per second)	110, 150, 300, 600, 1200, 2400, 4800, 9600, or 19,200	None
parity	N (none) O (odd) E (even)	E
data bits	7 or 8	7
stop bits	1 or 2	2 (if baud = 110)
		1 (if baud not = 110)

2. The P parameter. The P parameter indicates that the asynchronous adapter is being used for a serial interface printer.

   If you specify P in the command format 3, MODE continuously retries time-out errors.

To stop the retry loop press **Ctrl–Break**.

To stop MODE from continuously retrying time–out errors, reenter the MODE command without the P parameter.

Examples

```
MODE COM1:12,N,8,1,P
```

Sets the mode of operation for adapter COM1 as follows:

- 1200 baud rate
- no parity
- 8 data bits
- 1 stop bit.

To use the default values, enter the following command:

```
MODE COM1:12,,,,P
```

The parity defaults to even, the data bits default to seven, and the stop bits default to one (see Table 5–7).

FUNCTION 4: REDIRECTING PARALLEL PRINTER OUTPUT TO AN ASYNCHRONOUS COMMUNICATIONS ADAPTER

Syntax

MODE LPT*n*[:]=COM*m*

Comments

n specifies the printer; use 1, 2, or 3.

m specifies the asynchronous communication adapter number; use 1, 2, 3, or 4.

This command format causes DOS to redirect output for printer LPT*n* to the asynchronous adapter COM*m*. Before you can use MODE to redirect parallel printer output to a serial (asynchronous) device, you must specify the asynchronous communication modes by using the command format found in FUNCTION 3 as follows:

MODE COM*m*[:]*baud*[,*parity*[,*databits*[,*stopbits*[,:]]]]

If the serial device is a printer, include the P parameter in the command.

The command

MODE LPT*n*

disables any redirection of LPT*n*.

Note

MODE makes the printer and the asynchronous communications adapter code resident in memory. This code increases the size of DOS in memory.

FUNCTION 5: SETTING OR DISPLAYING CODE PAGES FOR PARALLEL PRINTERS OR YOUR CONSOLE SCREEN DEVICE

Syntax

```
PREPARE=[[yyy][drive:][path]filename]
   and
MODE DEVICE CODE PAGE SELECT=yyy
MODE DEVICE CODE PAGE REFRESH
MODE DEVICE CODE PAGE [/status]
```

Options

device specifies the device to support code page switching. Valid device names are CON, LPT1, LPT2, and LPT3.

yyy specifies a code page. Valid code pages are 437, 850, 860,863, and 865.

filename identifies the name of the Code Page Information (.CPI) file MS-DOS should use to prepare a code page for the devices pecified.

There are four keywords that you can use with the MODE DEVICE CODE PAGE command. Each causes the mode command to perform a different function. The following table explains each keyword:

Keyword	Function
PREPARE	Tells MS-DOS to prepare code pages for a given device. You must prepare a code page for a device before you can use it with that device.
SELECT	Specifies which code page you want to use with a device. You must prepare a code page before you can select it.
REFRESH	If the prepared code pages for a device are lost due to hardware or other error, this keyword reinstates the prepared code pages.

/status	Displays the current code pages prepared and/or selected for a device. Note that the following commands both produce the same results: MODE CON CODE PAGE MODE CON CODE PAGE /status Typing /status is optional.

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

Notes

You can use the following abbreviations with the MODE command for code page modes:

CP	CODE PAGE
/sta	/status
PREP	PREPARE
SEL	SELECT
FEF	REFRESH

If you want to print files whenever you start MS–DOS, include MODE commands in your AUTOEXEC.BAT file.

Examples

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

```
MODE COM1:48,E,,,p

MODE LPT1:=COM1:
```

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

```
MODE LPT1:
```

This command disables any redirection of LPT1. Suppose you want your computer to print on a printer connected to your computer's second parallel printer port (LPT2). If you want to print with 80 characters per line and 8 characters per inch, you would type

```
MODE LPT2: 80,8

        or

MODE LPT2:,8
```

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

```
MODE LPT2:80,8,P
```

To stop retrying to print, you can press **CONTROL-BREAK** or type the MODE command without the P option.

MORE

E

Purpose

Sends output to the console one screen at a time.

Syntax

MORE

Comments

MORE is a filter that reads from standard input (for instance, a command from your terminal) and displays one screen of information at a time. The MORE command then pauses and displays the "--More--" message at the bottom of your screen. To continue displaying information, press **Enter** and keep pressing it until you have read all the data.

To hold input information until it is displayed, the MORE command creates a temporary file on the disk. If the disk is full or write–protected, however, the MORE command will not work.

Examples

MORE is useful for viewing long files. For example, if you have a long file of customers, you could use the MORE command to view it one screen at a time. Suppose this file is called CLIENTS.NEW. To see it you would just type the following command:

```
TYPE CLIENTS.NEW | MORE
```

You can also redirect input from a file to more, for example:

```
MORE < CLIENTS.NEW
```

This command also sends the file CLIENTS.NEW to the screen one screenful at a time.

NLSFUNC

E

Purpose

Loads country-specific information.

Syntax

NLSFUNC[[*drive*:][*path*]*filename*]

Comments

The *filename* parameter specifies the file containing country-specific information.

The NLSFUNC command supports the use of extended country-specific information and code page switching.

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

Examples

Suppose you have a file on your disk called NEWCDPG.SYS that contains country-specific information. If you want to use the information from that file rather than the COUNTRY.SYS file, you would type the following command:

```
NLSFUNC NEWCDPG.SYS
```

To use the default country-specific information found in the COUNTRY.SYS file, simply type this command:

```
NLSFUNC
```

PATH

Purpose

Sets a command search path.

Syntax

PATH [*drive*:][*path*][;[*drive*:][*path*]...]

or

PATH ;

Comments

The PATH command lets you tell MS–DOS which directories to search for external commands--after it searches your working directory. The default value is no path.

For instance, to tell MS–DOS to search the \USER\PETE directory for external commands, you would simply type the PATH command followed by the directory name \USER\PETE. Then, until you exit MS–DOS or set another path, MS–DOS searches the \USER\PETE directory for external commands.

You can tell MS–DOS to search more than one path by specifying several paths separated by semicolons. If you use the PATH command without options, it prints the current path. And if you use the following command, MS–DOS sets the NUL path:

```
PATH ;
```

This command means that MS–DOS searches only the working directory for external commands.

Example

The following command tells MS–DOS to search three directories to find external commands (the three paths for these directories are \USER\PETE, B:\USER\EMILY, and \BIN):

```
PATH \USER\PETE;B:\USER\EMILY;\BIN
```

MS–DOS searches the paths in the order specified in the PATH command.

PRINT

E

Purpose

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

Syntax

PRINT[/D:*device*] [/B:*size*] [/U:*value1*] [/M:*value2*] [/S:*time–slice*]
[/Q:*qsize*] [/T] [/C] [/P] [*drive*:] [*pathname*]

Comments

You can use the PRINT command only if you have a lineprinter attached to your computer. If you do have a lineprinter you can use the following switches with this command:

/U:value — Specifies the number of clock ticks print will wait for a printer. If the printer is not available within the time specified, the job will not run. The default for value is 1.

/M:value — Specifies the number of clock ticks print can take to print a character on the printer. Valid values for value 2 range from 1 to 255. The default is 2.

/S:times — The interval of time to be used by the MS–DOS scheduler for the PRINT command.

/Q:qsize — Specifies the number of files allowed in the print queue — if you want more than 10. The minimum value for the /Q switch is 4, the maximum, 32, and the default, 10. To change this default number of files, you must use the PRINT command without any filenames; for example,

```
PRINT /Q:32
```

/T — Deletes all files in the print queue (those files waiting to be printed).

/C — Turns on cancel mode and removes the preceding filename and all following filenames from the print queue.

/P — Turns on print mode and adds the preceding filename and all following filenames to the print queue.

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

Notes

The /D, /B, /U, /M, /S, and /Q switches are allowed only the first time you run the PRINT command after starting MS–DOS.

Each print queue entry may contain a maximum of 64 characters, including the drive name. So you may need to change directories first to avoid using extensive pathnames.

Some applications have their own print commands. You should use the application's print facility to print files that you create with the application.

PROMPT

Purpose

Changes the MS-DOS command prompt.

Syntax

PROMPT [[*text*][*$character*]...]

Comments

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

You can use the following characters in the prompt command to create special prompts. You must precede each character with a dollar sign ($):

Specify this character	To get this prompt
$	The $ character
t	The current time
d	The current date
p	The working directory of the default drive
v	The version number
n	The default drive
g	The > character
l	The < character
b	The | character
_	Enter-LINEFEED
s	A space (leading only)
e	ASCII code X'1B' (escape)

Examples

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

```
PROMPT $p
```

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

```
Time = (current time)
Date = (current date)
```

```
PROMPT time = $t$ date = $d
```

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

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

RECOVER

E ~~NET~~

Purpose

Recovers a file or disk containing bad sectors.

Syntax

RECOVER [*drive*:]

or

RECOVER *drive*:[*pathname*]

Comments

If the CHKDSK command shows that a sector on your disk is bad, you can use thẹ RECOVER command to recover the entire disk or just the file containing the bad sector.

This action causes MS–DOS to read the file sector by sector and to skip the bad sectors. When MS–DOS finds a bad sector, it marks the sector so that it no longer allocates your data to that sector.

Examples

To recover a disk in drive A you would use the following command:

```
    RECOVER A:
```

Suppose you have a file named PENCIL.AD that has a few bad sectors. To recover this file you would use the following command:

```
    RECOVER PENCIL.AD
```

REN (RENAME)

Synonym

RENAME

Purpose

Changes the name of a file.

Syntax

REN [*drive*:]*pathname pathname*

Comments

The REN command renames all files matching the first *pathname*. However, because you cannot rename files across disk drives, the REN command ignores any drive name that you specify with the second *pathname*.

You may use wildcards (* or ?) in either *pathname* option, but if you use them in the second *pathname*, REN will not change the positions of the corresponding character.

Examples

The following command changes the extension of all filenames ending in .TXT to .DOC:

```
REN *.TXT *.DOC
```

In the next example, REN renames a file named CHAP10 (on drive B) to PART10:

```
REN B:CHAP10 PART10
```

The newly renamed file PART10 remains on drive B.

REPLACE

E

Purpose

The REPLACE command allows you to selectively add or replace files on a destination disk with files from a source disk. When replacing files, the source and destination files must have the same name.

Syntax

REPLACE [*drive*:] [*path*]*filename*[*.ext*] [*drive*:] [*path*] [/A] [/P] [/R] [/S] [/W]

Comments

[*drive*:] [*path*]*filename*[*.ext*] specifies files on the source disk that are to be added or replaced on the destination. This file specification can be up to 63 characters long.

The second [*drive*:] [*path*] specifies the drive and path of the destination.

Options

/A copies specified files from the source that do not already exist on the destination. /A cannot be used with /S.

/P displays a prompt before each specified file found on the destination is added or replaced.

/R replaces all specified files on the destination.

/S searches all the directories on the destination for the files that match the source file. /A cannot be used with /S.

/W waits for you to insert a diskette before searching for source files. If /W is not used, the search begins immediately.

Example

The following command replaces all files named README.DOC on hard disk C: with the version of README.DOC in subdirectory C: \UPDATES.

```
REPLACE C:\UPDATES\README.DOC C:\ /S
```

RESTORE

Purpose

Restores files that were backed up using the backup program.

Syntax

RESTORE *drive1:* [*drive2:*][*pathname*]
[/S][/P][/B:*date*][/A:*date*][/E:*time*][/L:*time*][/M] [/N]

Switches

/B:date Restores only those files last modified on or before date.

/A:date Restores only those files last modified on or after date.

/E:time Restores only those files last modified at or earlier than time.

/L:time Restores only those files last modified at or later than time.

/M Restores only those files modified since the last backup.

/N Restores only those files that no longer exist on the target disk.

Once MS–DOS has restored the file, use the DIR or TYPE command to make sure that the file was restored properly.

Notes

RESTORE cannot restore the system files. Use the SYS command to restore these files.

The MS–DOS 3.3 RESTORE command will restore files backed up with either the MS–DOS 3.3 BACKUP command, or an earlier version of BACKUP. Upon completion, RESTORE returns one of the following exit codes:

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

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

Example

To restore the file INVEST.MNT from the backup disk in drive A to the \IRSHARPE directory on drive C, type thefollowing:

```
RESTORE A: C:\IRSHARPE\INVEST.MNT
```

Press the **ENTER** key to let MS–DOS know that the backup disk is in drive A. Then once MS–DOS has restored the file, use the DIR or TYPE command to make sure that the file was restored properly.

E

RETRACT

Purpose

The RETRACT command moves the heads on all installed Hard Disk units to a safe, non–data landing zone. This will help protect the media and heads from accidental damage during vibration, moving, or shipping of the computer.

Syntax

RETRACT

Comments

This utility may be run (without risk) each time you turn off the computer. This will provide greater protection from vibration or shock which may occur while the unit is not being used.

There are no options, and this utility will perform no function if there are no hard disk units present.

RMDIR (REMOVE DIRECTORY)

I

Synonym

RD

Purpose

Removes a directory from a multilevel directory structure.

Syntax

RMDIR [*drive*:]*path*

Comments

This command removes a directory that is empty except for the "." and ".." shorthand symbols. Before you can remove a directory entirely, you must delete its files and subdirectories.

Example

Suppose you want to remove a directory named \USER\PETE. You first issue a DIR command for the \USER\PETE path to ensure that the directory is empty; you then type the following command:

```
RMDIR \USER\PETE
```

SELECT

Purpose

The SELECT command creates a version of DOS on the destination diskette with the keyboard layout and date and time format of your choice.

Syntax

SELECT [[A:|B:] D:[*path*]] *xxx yy*

Comments

[A:|B: specifies the source drive and path. If the source drive is specified, a destination drive must also be specified. The default source drive is A:.

[D:[*path*]] specifies the destination drive and path. The destination and source drives cannot be the same. The default destination drive is B:. The default path is the destination drive's root directory.

xxx specifies the country code. The country code sets the date and time format, the currency symbol, and the decimal separator.

yy specifies the keyboard code. The keyboard code sets the keyboard layout. Refer to Appendix B for examples of available keyboard layouts.

SELECT uses FORMAT and XCOPY to make a copy of DOS in the destination drive, and creates the following two files.

- CONFIG.SYS file that contains COUNTRY=command
- AUTOEXEC.BAT file that contains the KEYB command, except in the case of a U.S. keyboard.

CAUTION

Since SELECT uses the FORMAT command to format, all data on the destination disk is destroyed. Before any data is erased, SELECT prompts you to confirm that you want to continue.

If you are formatting a hard disk, SELECT prompts you for a volume label.

All paths must be specified from the root directory. Do not use the . or .. entries to specify paths.

SELECT boots the new keyboard through the AUTOEXEC.BAT file. See Section 5 for more information on AUTOEXEC.BAT files.

The table on page E–2 lists country codes and keyboard codes for countries that SELECT supports.

SET

Purpose

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

Syntax

SET [*string*=[*string*]]

Comments

You should use the SET command only if you want to set values for programs you have written.

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

If you specify just the first *string*, SET removes any previous setting of that string from the environment. Or if you use the set command without options, MS–DOS displays the current environment settings.

When batch processing, you can also use the SET command to define your replaceable parameters by name instead of by number. For example, if your batch file contains the statement "TYPE %file%", you could use the SET command to set the name that MS–DOS will use for that variable. In the following command, for example, SET replaces the %file% parameter with the filename TAXES.86:

```
SET FILE=TAXES.86
```

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

The SET command is especially useful in the AUTOEXEC.BAT file, because it lets you automatically set strings or parameters when you start MS–DOS. See Section 5, "Batch Processing," for more information about the AUTOEXEC.BAT file.

Example

The following command sets the string "include" to C:\INC until you change it with another set command:

```
SET INCLUDE=C:\INC
```

If you just type the SET command by itself, MS–DOS displays the current environment settings.

SHARE

E

Purpose

Installs file sharing and locking.

Syntax

SHARE [/F:*space*] [/L:*locks*]

Comments

You can see the SHARE command only when networking is active. If you want to install shared files, you can include the SHARE command in your AUTOEXEC.BAT file. To learn more about shared files, see the *Microsoft Networks Manager's Guide*.

MS–DOS has a storage area that it uses to record file sharing information; to allocate file space (in bytes) for this area, you use the /F:*space* switch.

Each open file requires enough space for the length of the full filename plus 11 bytes, since the average pathname is around 20 bytes in length. The default value for the /F switch is 2048.

The /L:*locks* switch allocates the number of locks you want to allow. The default value for the /L switch is 20.

Once you have used the SHARE command in an MS–DOS session, all read and write requests are checked by MS–DOS.

Example

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

```
SHARE
```

SORT

Purpose

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

Syntax

SORT [*drive*:][*pathname*][/R][/+*n*]

Comments

The SORT command lets you alphabetize a file according to the character in a certain column. You specify the file by the *drive*: and *pathname* options. The two other SORT options, the /R and /+*n* switches, are described as follows:

/R Reverses the sort, that is, sorts from Z to A.

/+*n* Sorts the file according to the character in column *n*, where n is some number. If you do not specify this switch, the SORT command sorts the file according to the character in the first column.

SORT does not distinguish between uppercase and lowercase letters.

Examples

The following command reads the file EXPENSES.TXT, sorts it in reverse order, and writes the output to a file named BUDGET.TXT:

```
SORT /R EXPENSES.TXT BUDGET.TXT
```

The following command pipes the output of the DIR command to the SORT filter. This filter sorts the directory listing starting with column 14 (the column in the directory listing that contains the file size) and sends the output to the screen. The result is a directory, sorted by file size:

```
DIR | SORT /+14
```

The following command does the same thing as the previous one, except that the MORE filter gives you a chance to read the sorted directory one screen at a time:

```
DIR | SORT /+14 | MORE
```

SUBST

E NET

Purpose

Substitutes a path with a drive letter.

Syntax

SUBST [*drive*:] [*path*] [/D]

Comments

The SUBST command lets you associate a *path* with a drive letter. This drive letter then represents a *virtual drive* because you can use the drive letter in commands as if it represented an actual drive.

When MS–DOS finds a command that uses a virtual drive, it replaces the drive letter with the path.

If you type the SUBST command by itself, MS–DOS displays the names of the virtual drives in effect.

To delete a virtual drive you use the /D switch.

Example

The following command creates a virtual drive, Z, for the pathname B:\USER\BETTY\FORMS:

```
SUBST Z: B:\USER\BETTY\FORMS
```

Now, instead of typing the full pathname, you can get to this directory by simply typing the name of the virtual drive:

```
Z:
```

SYS

E NET

Purpose

The SYS function transfers MS–DOS system files from the disk in the default drive to the disk in the destination drive.

Syntax

SYS *drive:*

Comments

drive: specifies the destination drive.

Use SYS to transfer MSDOS.SYS and IO.SYS files to update your system and to transfer MS–DOS system files to application program diskettes that use MS–DOS.

SYS does not copy the COMMAND.COM file; you must copy COMMAND.COM into the root directory.

If the destination disk contains no system files, it must either be empty or be formatted with the FORMAT *drive*:/S or FORMAT *drive*:/B command. See the FORMAT command, described elsewhere in this section, for information on preparing disks to use MS–DOS.

If the destination disk contains IO.SYS and MSDOS.SYS files, they must be the same size as the system files you want to transfer.

SYS does not work on a network drive.

Example

If you want to display the contents of a file called HOLIDAY.MAR you would type the following command:

```
TYPE HOLIDAY.MAR
```

TIME

Purpose

Displays and sets the time.

Syntax

TIME [*hours:minutes*[*:seconds*[*:hundredths*]]]

Comments

Time is entered in a 24–hour clock format. If you just type the TIME command by itself, the following message is displayed:

```
Current time is hh:mm:ss.cc
Enter new time:
```

If you don't want to change the time shown, you simply press **Enter**. And if you want to change the time after you have started MS–DOS (for example, to 8:20 a.m.), type the time command followed by 8:20 in response to the MS–DOS prompt. Note that letters are not allowed; instead, you must type the time using numbers only. The allowed options are:

hours = 0–24
minutes = 0–59
seconds = 0–59
hundredths = 0–99

Separate the hour and minute entries by a colon. You do not have to type the *ss* (seconds) or *cc* (hundredths of a second).

Example

If you do not type a valid time, MS–DOS displays the following message and then waits for you to type a valid time:

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

As with the DATE command, you can change the TIME command format by changing the COUNTRY command in the CONFIG.SYS file. See Appendix C, "How to Configure Your System," for more information.

TREE

Purpose

The TREE command displays all the directory paths found on the specified drive, and optionally lists the files in each subdirectory.

TREE displays the full pathname and names of all directories defined within the path.

Syntax

TREE [*drive*:][/F]

Comments

[*drive*:] specifies the drive containing directories you want to examine. If you do not specify a drive, TREE uses the default drive.

Options

The /F switch displays the names of all files in each subdirectory. /F does not cause TREE to list files in the root directory.

Examples

Figure 4–1 presents a typical directory structure and describes the output that TREE would display for this directory. This figure represents the directory structure for a disk with the label MYDISK on drive A.

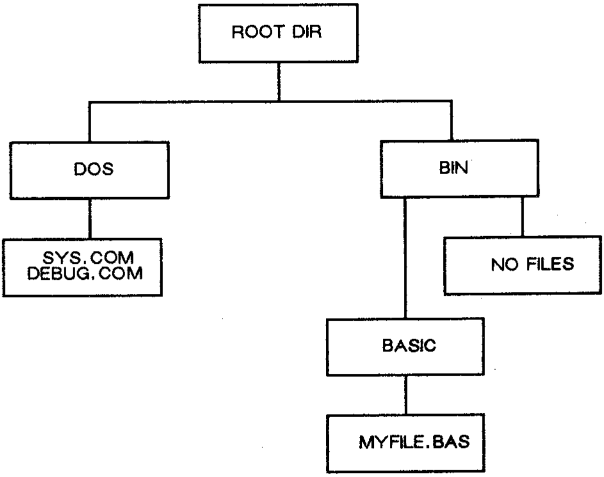
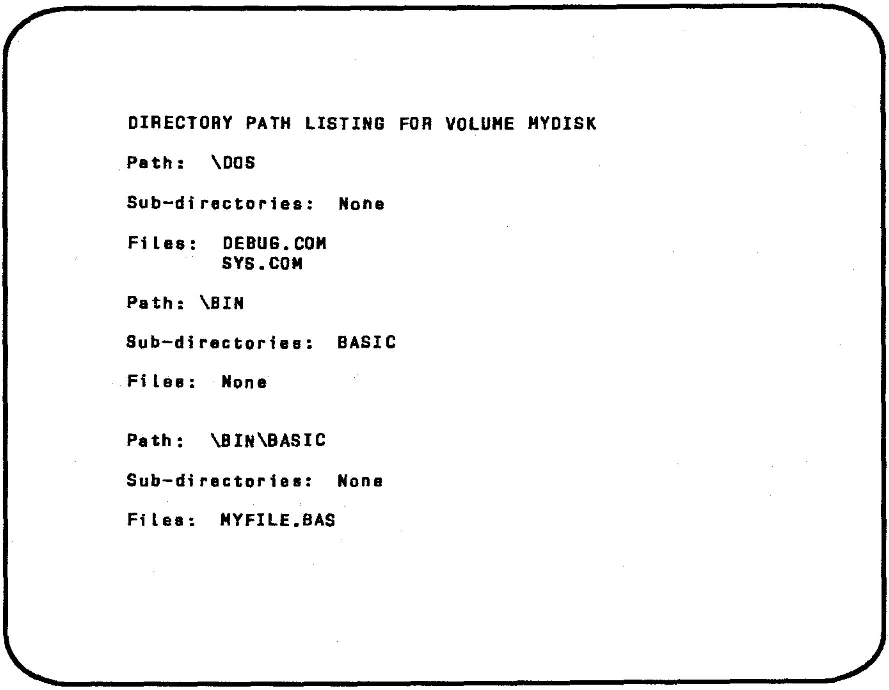


Figure 4–1 TREE Directory Path Listing

The command

TREE /F

displays the following output on your screen.



```
DIRECTORY PATH LISTING FOR VOLUME MYDISK

Path:  \DOS

Sub-directories:  None

Files:  DEBUG.COM
        SYS.COM

Path: \BIN

Sub-directories:  BASIC

Files:  None


Path:  \BIN\BASIC

Sub-directories:  None

Files:  MYFILE.BAS
```

The following examples show that you can redirect the output from TREE to a file or printer by using the pipe symbol >.

The following command causes TREE to list all directories on drive B. MS-DOS redirects the output to the file MYFILE.LST in the current directory.

TREE B:/F > MYFILE.LST

MYFILE.LST contains the names of all subdirectories and files for each directory level.

The following command sends all the subdirectories and filenames from drive A to the printer:

TREE A:/F > PRN

E

TURBO

Purpose

The TURBO command sets the processor speed to high speed or low speed.

Syntax

```
TURBO [+|-]
```

Comments

The command TURBO + sets the processor speed to high. TURBO – sets it to low.

The TURBO command without an option displays a description of the TURBO command; the processor speed is not changed.

Note that speed change occurs only on PowerMates with multi–speed capacity; no speed change occurs on single–speed models.

TYPE

Purpose

Displays the contents of a text file on the screen.

Syntax

TYPE [*drive*:]*filename*

Comments

To view a text file without modifying it, you can use the TYPE command. (Use DIR to find the name of a file, and EDLIN to change the contents of a file.)

Note that when you use TYPE to display a file that contains tabs, all the tabs are expanded to 8 spaces wide. Also, if you try to display a binary file, you may see strange characters on the screen, including bells, formfeeds, and escape sequences.

Example

If you want to display the contents of a file called HOLIDAY.MAR you would type the following command:

```
TYPE HOLIDAY.MAR
```

VER

Purpose

Prints the MS–DOS version number.

Syntax

VER

Comment

If you want to know what version of MS–DOS you are using, you simply type the VER command. The version number will then be displayed on your screen.

Example

When you type the VER command, the following message is displayed:

```
    MS–DOS Version 3.20
```

VERIFY

I

Purpose

Turns the verify switch on or off when writing to a disk.

Syntax

VERIFY [ON]

or

VERIFY [OFF]

Comments

You can use this command to verify that your files are written correctly to the disk (no bad sectors, for example). MS–DOS performs a VERIFY each time you write data to a disk. You will receive an error message only if MS–DOS is unable to successfully write your data to a disk.

Examples

If you want to know the current setting of VERIFY, use the VERIFY command without an option:

```
VERIFY
```

VERIFY ON remains in effect until a program changes it (by a SET VERIFY system call), or until you type the following:

```
VERIFY OFF
```

This command has the same purpose as the /V switch in the COPY command.

VOL (VOLUME)

Purpose

Displays the disk volume label or volume ID, if it exists.

Syntax

VOL [*drive*:]

Comments

This command displays the volume label of the disk in the specified *drive*:. If you do not type a drive letter, MS–DOS displays the volume label of the disk in the default drive.

Example

If you want to see the volume label for a disk in drive A, you could type the following:

```
VOL A:
```

If the volume label is "DOS 3–2" MS–DOS responds by displaying the message:

```
Volume in drive A is DOS 3–2
```

XCOPY

E

Purpose

The XCOPY command selectively copies groups of files that can include subdirectories.

Syntax

XCOPY[*drive:*] [*path*]*filename*[.*ext*] [*drive:*] [*path*] [*filename*[.*ext*]] [/A] [/D] [/E] [/M] [/P] [/S] [/V] [/W]

or

XCOPY [*drive:*] *path*[*filename*[.*ext*]] [*drive:*] [*path*] [*filename*[.*ext*]] [/A] [/D] [/E] [/M] [/P] [/S] [/V] [/W]

or

XCOPY
drive:[*path*] [*filename*[.*ext*]] [*drive:*] [*path*] [*filename*[.*ext*]] [/A] [/D] [/E] [/M] [/P] [/S] [/V] [/W]

Comments

The first file specification indicates the starting point of the files to be copied. It can be any combination of the drive, file, or directory parameters. The second file specification indicates the destination drive, path, or filename for the files being copied.

Options

/A copies files whose archive bit is set. When the copy procedure is complete, XCOPY will not turn off the archive bit. An archive bit that is set indicates that the file has been created or changed since the last XCOPY /M.

/D copies files whose date is equal to or later than the date specified in the XCOPY command. The date format must match the date format selected in the SELECT or COUNTRY commands. For example, /D:05/15/86 copies files later than May 15, 1986.

/E creates empty subdirectories on the destination that may result from the copy procedure. If /E is not specified, these empty subdirectories are not created.

/M copies files whose archive bit is set. A set archive bit indicates that the file has been created or changed since the last XCOPY /M. /M is useful when using XCOPY to back up your files. When the copy procedure is complete, XCOPY /M turns off the archive bit.

/P prompts you before copying each file.

/S copies all files in and below the source directory. If /S is not used, only files within the source directory are copied.

/V verifies that all sectors written to the destination diskette are correct.

/W waits for you to insert a diskette before searching for the source files. If /W is not used, the search begins immediately.

Notes

The default start directory is the current directory. The default filename is *.*

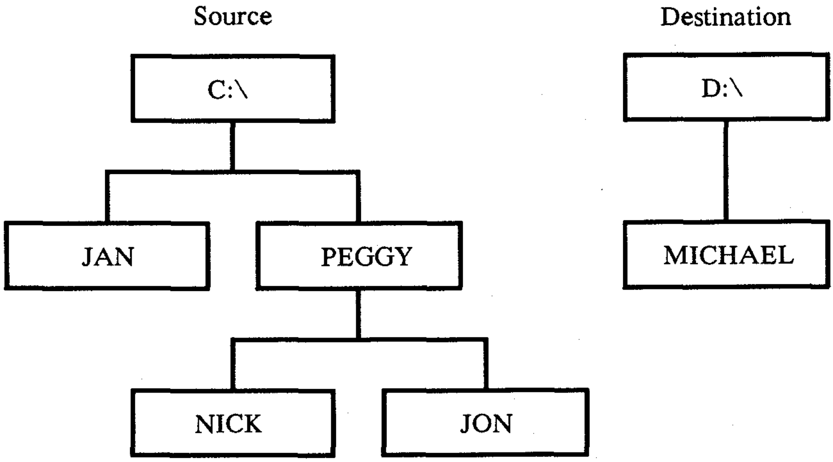
If the specified target path does not exist, the directories will be created before the copy procedure begins. The source or destination drive, path, and filename specification can be up to 63 characters long.

The reserved devices, CON and LPT1, cannot be specified as filenames with XCOPY.

Be sure that the destination space is large enough to hold the copied files. If the destination space is too small, the message "Disk full" appears and the program ends.

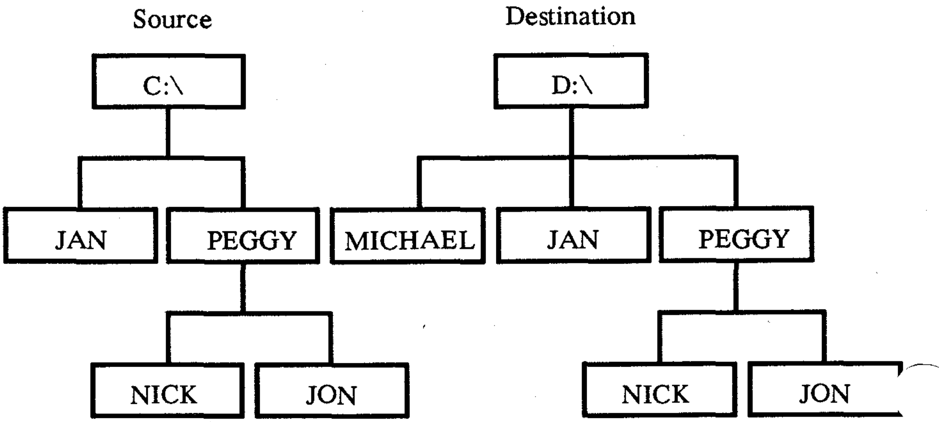
Examples

Assume that you have the following directory structures:



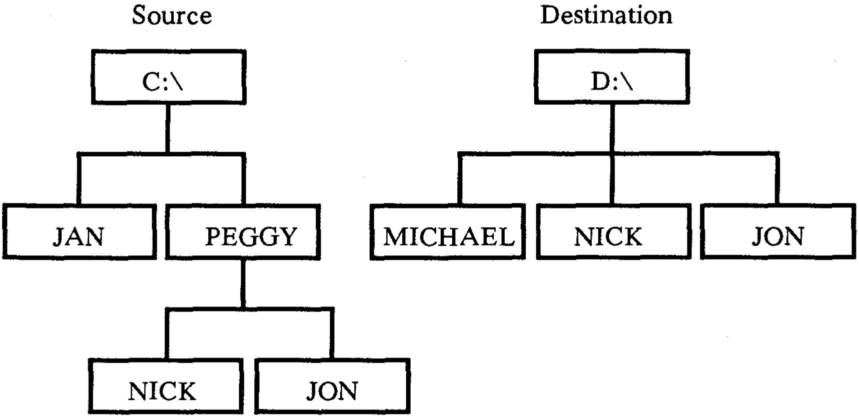
The command shown below results in the following directory structure:

```
XCOPY C:\ D:\ /S
```



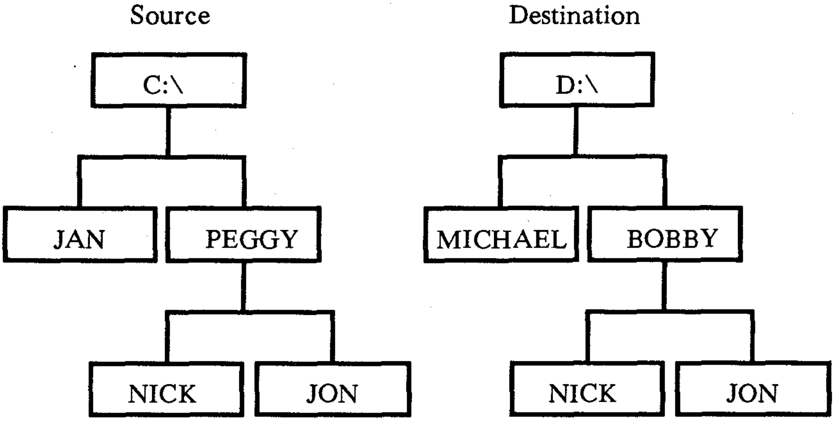
The command shown below results in the following directory structrure:

```
XCOPY C:\PEGGY D:\ /S
```



The command shown below results in the following directory structrure:

```
XCOPY C:\PEGGY D:\BOBBY /S
```



The command shown below is invalid because it involves a cyclic copy:

```
    XCOPY C:\ C:\PEGGY /S
```

The starting source directory cannot be a member of the parents of the destination starting directory. The message "Cannot perform a cyclic copy" is displayed.

Section 5

Batch Processing

In this chapter you will learn:

- how to create a batch file
- how an AUTOEXEC.BAT file works
- how to use replaceable parameters in a batch file
- how to run a batch file
- how to do multitasking with batch files.

If you are not writing batch programs you do not need to read this chapter.

WHY USE BATCH FILES?

You may often find yourself repeatedly typing the same sequence of commands to perform some common task. With MS-DOS you can put this command sequence into a special file called a *batch file*, and then run the whole sequence of commands by simply typing the name of the batch file. Note that you don't need to type the batch file's extension, even though all your batch files must include the .BAT extension in their filenames.

MS-DOS performs these "batches" of your commands just as if you had typed them from the keyboard. This is called *batch processing*. By using a batch file, you only have to remember to type one command, instead of several. In effect, you use batch files to create personalized commands.

HOW TO CREATE BATCH FILES

You can create a batch file by using EDLIN, the MS-DOS line editor, or by using the COPY command. If you want to create files with EDLIN, you should refer to Sections 7 and 8 for more information. The examples in this section show you how to use the COPY command to create batch files.

Suppose, for example, that you want to create a batch file to format and check a new disk. To do this you simply follow these steps:

1. First, type the following:

   ```
   COPY CON CHECKNEW.BAT
   ```

   Press **Enter**. This command tells MS–DOS to copy the information from the console (keyboard) to the file CHECKNEW.BAT.

2. Next, type the following lines, pressing **Enter** after each:

   ```
   rem This is a file to format and
   rem check new disks.
   rem It is named CHECKNEW.BAT.
   pause Insert new disk in drive B:
   FORMAT B: /V
   CHKDSK B:
   ```

3. After the last line, press **Ctrl–Z** and then press **Enter** to save the batch file. MS–DOS displays the message "1 File(s) copied" to show that it created the file.

4. Now, to execute the file, simply type the following command:

   ```
   CHECKNEW
   ```

The result is the same as if the lines in the .BAT file were entered from the keyboard as individual commands.

Here are a few things you should know before you run a batch process with MS–DOS:

- You must name each batch file with an extension of .BAT. To execute a batch file, you type only its filename and not the extension.
- If you press **Ctrl–C** while the batch file is running, MS–DOS asks you to confirm that you want to terminate the batch process.
- If you remove the disk that contains a batch file being run, MS–DOS prompts you to reinsert the disk so that it can continue processing the file.
- You can specify the name of another batch file as the last command in a batch file. This feature allows you to call one batch file from another when the first has finished.
- You can use any of the redirection symbols (< > >>) in a batch file. See "REDIRECTING COMMAND INPUT AND OUTPUT" in Section 2 for more information on using these symbols.

- You may use an @ character at the front of a command line in a batch file to prevent that line from echoing.
- You can use the pipe symbol (|) in a batch file.
- Setting the directory or drive affects every subsequent command in the batch file.
- Setting environment strings also affects every subsequent command in the batch file.

If you have more than one external command with the same name, MS-DOS will run only one of them, according to the following order of precedence: .COM, .EXE, .BAT.

Suppose, for example, that your disk includes the files FORMAT.EXE and FORMAT.BAT. If you were to type the external command FORMAT, MS-DOS would always run the program FORMAT.EXE first. In order to run the batch file FORMAT.BAT, you would have to place it in a separate directory and give a path along with the external command.

WHAT IS AN AUTOEXEC.BAT FILE?

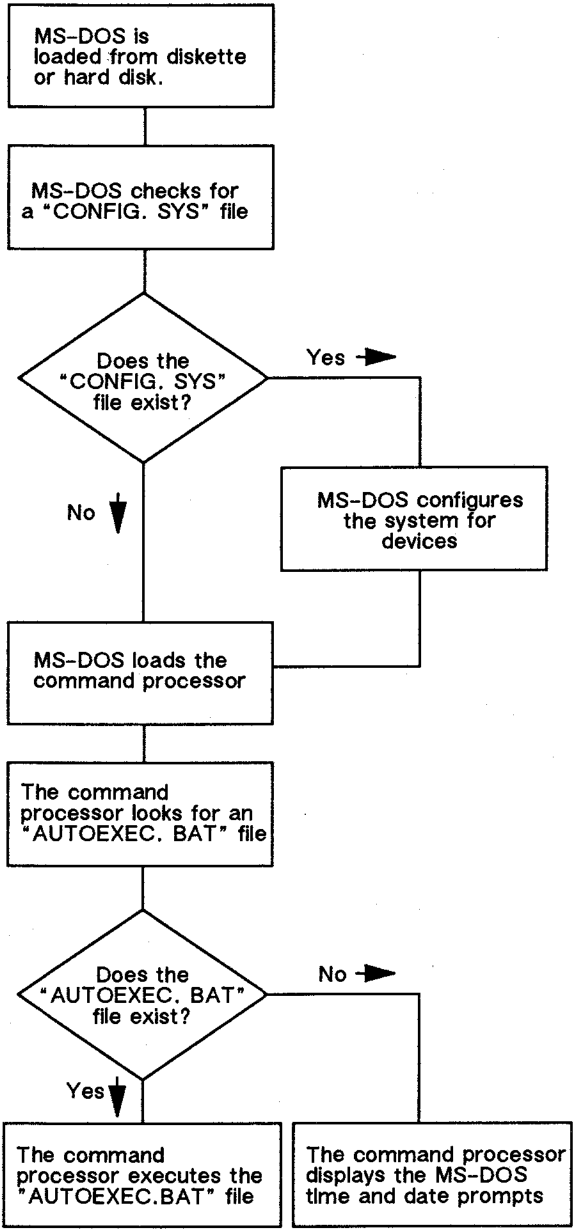
An AUTOEXEC.BAT file lets you run programs automatically when you start MS-DOS. This can be useful when you want to run a specific application under MS-DOS, and when you want MS-DOS to execute a batch program each time you start your computer. By using an AUTOEXEC.BAT file you can avoid loading two separate disks just to perform these tasks.

When you start your computer, MS-DOS searches the root directory of the default disk drive for a file named AUTOEXEC.BAT. If it finds the AUTOEXEC.BAT file, MS-DOS immediately processes it, bypassing the date and time prompts. If MS-DOS does not find an AUTOEXEC.BAT file, then the DATE and TIME prompts appear automatically.

MS-DOS does not prompt you for a current date and time unless you include the DATE and TIME commands in your AUTOEXEC.BAT file.

It's a good idea to add these two commands to your AUTOEXEC.BAT file, since MS-DOS uses this information to keep your directory current. See Section 4, "MS-DOS Commands," for more information on the DATE and TIME commands.

The following flowchart shows what happens when you start MS-DOS:


MS-DOS is loaded from diskette or hard disk.
MS-DOS checks for a "CONFIG. SYS" file
Does the "CONFIG. SYS" file exist?
Yes
No
MS-DOS configures the system for devices
MS-DOS loads the command processor
The command processor looks for an "AUTOEXEC. BAT" file
Does the "AUTOEXEC. BAT" file exist?
No
Yes
The command processor executes the "AUTOEXEC.BAT" file
The command processor displays the MS-DOS time and date prompts

HOW TO CREATE AN AUTOEXEC.BAT FILE

There are many things you can do with an AUTOEXEC.BAT file to help you use MS–DOS more efficiently. For instance, you will probably want to set the time and date, your path, and any other options that you plan to use on a regular basis.

If, for example, you want to automatically load GW–BASIC and run a program called menu each time you start MS–DOS, you could create an AUTOEXEC.BAT file as follows:

1. Type the following command and then press **Enter**:

   ```
   COPY CON AUTOEXEC.BAT
   ```

   This command tells MS–DOS to copy what you type from the keyboard into the AUTOEXEC.BAT file. Note that you must put the AUTOEXEC.BAT file in the root directory of your MS–DOS disk.

2. Now type the following lines:

   ```
   DATE
   TIME
   PATH=C:\;C:\BIN;A:\
   PROMPT [$p]
   CLS
   GWBASIC MENU
   ```

3. After the last line, press **Ctrl–Z** and press **Enter** to copy these lines into the AUTOEXEC.BAT file.

4. The MENU program will now run automatically whenever you start MS–DOS.

Once your AUTOEXEC.BAT file is set up as above, it will perform the following actions when you start MS–DOS: it will ask you to enter the date and time; it will set your command search path; and it will set your prompt to display the default drive and directory.

Finally, the AUTOEXEC.BAT file will clear the screen and tell MS–DOS to load GW–BASIC and run the MENU program. To run your own GW–BASIC program, type its name in place of MENU in the example. In addition to GW–BASIC programs, you can also put any MS–DOS command or series of commands in the AUTOEXEC.BAT file.

HOW TO CREATE A BATCH FILE WITH REPLACEABLE PARAMETERS

There may be times when you want to create a program and run it with different sets of data. These data may be stored in various MS-DOS files.

With MS-DOS you can create a batch (.BAT) file with replaceable (dummy) parameters, where a *parameter* is a command option that you define. These parameters, named %0 through %9, hold the places for the values that you supply when you give the batch command.

Replaceable parameters make batch files flexible and easy to use. For example, you can create a batch file called SORTER.BAT that sorts a file containing a specific sequence of characters or strings. Each time you execute the SORTER batch file, you tell MS-DOS which string you want, which file to search to find that string, and which temporary file to use for sorting. SORTER would then print the resulting list on the printer.

1. To create the SORTER.BAT file, type the following command and then press the **Enter** key:

   ```
   CON SORTER.BAT
   ```

2. Now type the following lines:

   ```
   TYPE %2 | FIND "%1" > %3
   TYPE %3 | SORT > PRN
   DEL %3
   ```

3. To save the batch file, press **Ctrl-Z** and then **Enter**.

   The batch file SORTER.BAT now consists of three command lines and is on the disk in the default drive.

When you execute the file, MS-DOS sequentially replaces %1, %2, and %3 with the parameters you supply. If you use the dummy parameter %0, MS-DOS always replaces it with the drive name (if specified) and the filename of the batch file (for example, SORTER).

You can specify up to ten replaceable parameters (%0–%9). If you want to specify more than ten, refer to the SHIFT command later in this section.

If you use the percent sign as part of a filename within a batch file, you must type it twice. For example, to specify the file ABC%.EXE, you must type it as ABC%%.EXE in the batch file.

HOW TO RUN A BATCH FILE

To run the batch file SORTER.BAT, type the batch filename followed by the parameters that you want MS-DOS to substitute for %1, %2, and %3.

Suppose that on the disk in drive A you have a file that lists your customers' names and regions. The file might look something like this:

```
Shores, Sandy          north
Poster, Emily          south
Sharpe, Isabel R.      north
Fisher, Pete           east
Conrad, Betty          south
Rey, Fernando          north
Shaw, Rick             west
Moss, Chris            north
```

If you want to print an alphabetical list of the customers in the north, you can run the SORTER batch file, with the appropriate parameters, by typing the following command and then pressing **Enter**.

```
SORTER NORTH A:CUSTOMER TEMP.FIL
```

The output on the printer should look like this:

```
Moss, Chris            north
Rey, Fernando          north
Sharpe, Isabel R.      north
Shores, Sandy          north
```

The following table shows how MS-DOS replaces each of the parameters in the previous example:

Batch filename	(%0) sorter
Parameter1	(%1) north
Parameter2	(%2) a:customer
Parameter3	(%3) temp.fil

The result is the same as if you had typed each of the commands in SORTER with its parameters, as follows:

```
TYPE A:CUSTOMER | FIND "NORTH" > TEMP.FIL
TYPE TEMP.FIL | SORT > PRN
DEL TEMP.FIL
```

Using the batch file, however, saves typing time and is much easier to remember.

How to Use Temporary Files

When using batch files, you may often want to use a temporary file to hold your work. You could use the same name each time you wanted to use a temporary file.

However, if you are using more than one batch file that uses the same temporary file, you might lose the contents of this temporary file. To avoid this problem, you should use a replaceable parameter to specify the name of the temporary file. Then each time you run the batch file, you'll be able to substitute a unique filename and you won't have to worry about information from one batch file getting into another.

It's also a good idea to delete temporary files once you finish using them. Otherwise, these files would eventually take up all the space on your disk.

BATCH PROCESSING COMMANDS

Now that you have seen some of the capabilities of batch files, in this section you'll find out how to add power and flexibility to your batch programs by using batch processing commands. Table 5-1 lists these batch commands and describes what they do:

Table 5-1 Batch Processing Commands

COMMAND	WHAT IT DOES
CALL	Calls one batch file from another without ending itself.
ECHO	Turns the batch file echo feature on or off, or displays the current setting.
FOR	Performs a command for a set of files.
GOTO	Processes commands starting with the line after the specified label.
IF	Performs a command if a condition is met.
PAUSE	Pauses during the processing of a batch.
REM	Displays a comment in a batch file.
SHIFT	Increases the number of replaceable parameters in a batch process.

CALL

Purpose

Calls one batch file from another without ending itself.

Syntax

CALL [*drive*] [*path*] *batchfile* [*argument*]

Comments

Batchfile is the batch file you want to call.

Argument is the command in this batch file that will be run following *batchfile*.

The CALL command is used within one batch file to call another one. *Batchfile* must have a filename extension of .BAT. When *batchfile* terminates, the calling batch file resumes running at argument. If argument is omitted, the calling batch file resumes running at the command immediately following the CALL command.

Notes

Do not use pipes and redirection symbols with the CALL command.

A batch file can make a recursive call to itself, but there should be a termination condition that is eventually met.

Example

To run the CHECKNEW.BAT file from another batch file, you would use the following command within the first batch file:

```
CALL CHECKNEW
```

ECHO

I

Purpose

Turns the batch echo feature on nd off.

Syntax

ECHO [ON]

or

ECHO [OFF]

or

ECHO [*message*]

Comments

Normally, commands in a batch file are displayed ("echoed") on the screen when they are received by MS-DOS. You can turn off this feature by using the OFF option with the ECHO command. Similarly, you can turn the echo feature back on by using the ON option with ECHO.

If you do not specify ON or OFF, ECHO displays the current setting.

The command ECHO *message* (where message is a line of text) is only useful if ECHO is off and if you are using a batch file. If, in your batch file, you type the ECHO command followed by a message, you can print messages on your screen. You can also put several ECHO message commands in your batch file to display a message that is several lines in length.

Example

The following is an example of a batch file message of more than one line:

```
ECHO OFF
ECHO This batch file
ECHO formats and checks
ECHO new disks.
```

FOR

Purpose

Performs a command for a set of files.

Syntax

FOR *%%c* IN *set* DO *command*
(for batch processing)

FOR *%c* IN *set* DO *command*
(for interactive processing)

Comments

To avoid confusion with the %0–%9 batch parameters, the variable *c* can be any character except 0,1,2,3,...,9.

> _set_ is (_item_*)

This command sequentially sets the *%%c* variable to each member of *set*, and uses the variable to evaluate command. If a member of set is an expression involving a wildcard (* or ?), then the variable is set to each matching *item* from the disk. In this case, only one such item is in *set*, so the command ignores any item other than the first.

Examples

The following example binds the variable %f to files ending with *.ASM in the working directory.

```
FOR %%f IN (*.ASM ) DO MASM %%f
```

It then executes a command of the following form:

> MASM *filename*

filename could be any one of the following:

```
INVOICE.ASM
RECEIPTS.ASM
TAXES.ASM
```

The following example binds the variable %f to the files named REPORT, MEMO, and ADDRESS; it then deletes each of these files:

```
FOR %%f IN (REPORT MEMO ADDRESS) DO DEL %%f
```

You must use two percent signs (%%) so that one will remain after the batch parameter (%0–%9) processing is complete. If you had only *%f,* instead of *%%f,* then the batch parameter processor would see the %, look at *f,* decide that *%f* was an error (a bad parameter reference), and throw out the *%f* so that the FOR command would never see it.

If the FOR command is *not* in a batch file, you should use only one percent sign.

GOTO

Purpose

Processes commands starting with the line after the specified label.

Syntax

GOTO *label*

Comments

GOTO lets you take commands from the batch file beginning with the line after the *label*, where a label is defined as the characters following GOTO. This label may include spaces, but not other separators, such as semicolons or equal signs. If your batch file does not contain the *label*, the batch file terminates.

Any line in a batch file that starts with a colon (:) is ignored during batch processing.

Example

The following example sends the program processor to the label named *end* only if no errors occur when you format the disk in drive A:

```
:begin
ECHO OFF
FORMAT A: /S
IF errorlevel 0 GOTO end
ECHO An error occurred during formatting.
:end
ECHO End of batch file.
```

IF

Purpose

Performs a command based on the result of a condition.

Syntax

IF [NOT] *errorlevel number command*

or

IF [NOT] *string1* == *string2 command*

or

IF [NOT] EXIST *filename command*

Comments

The IF statement allows conditional execution of commands. When the condition is true, MS–DOS executes the *command*, otherwise it ignores the *command*.

The conditions are described as follows:

errorlevel number

> True if, and only if, the previous program executed by COMMAND.COM had an exit code of *number* or higher. (When a program finishes, it returns an exit code via MS–DOS.) You can use this condition to perform other tasks that are based on the previous program's exit code.

string1 == string2

> True if, and only if, *string1* and *string2* are identical after parameter substitution. Strings may not contain separators, such as commas, semicolons, equal signs, or spaces.

EXIST *filename*

> True if, and only if, *filename* exists.

If you specify the NOT parameter, MS–DOS executes the command when the condition is false.

Example

The following example prints the message "can't find datafile" if the file PRODUCT.DAT does not exist on the disk:

```
    IF NOT EXIST PRODUCT.DAT ECHO can't find datafile
```

PAUSE

Purpose

Suspends execution of a batch file.

Syntax

PAUSE [*comment*]

Comments

When a batch file is running, you may need to change disks or perform some other action. The PAUSE command suspends execution of the batch file until you press any key, unless you press the Ctrl–C key sequence.

When the command processor encounters PAUSE, it prints the following message:

```
Strike a key when ready . . .
```

If you press **Ctrl–C**, MS–DOS displays the following message:

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

If you type Y in response to this prompt, the batch file ends and control returns to the operating system. Therefore, you can use PAUSE to divide a batch file into pieces that allow you to end the batch command file at any intermediate point.

The *comment* parameter is useful when you want to display a special message. Unless ECHO is off, PAUSE displays this comment before the "Strike a key" message.

Note that the pause and comment line of your batch file will not appear if ECHO is off.

Example

Suppose you want a program to display a message that asks the user to change disks in one of the drives. To do this you might use the following command:

```
PAUSE Please put a new disk into drive A
```

If ECHO is on, this line will precede the "Strike a key" message when you run the batch file.

REM

I

Purpose

During execution of a batch file, REM displays remarks that are on the same line as the REM command in that batch file.

Syntax

REM [*comment*]

Comments

The *comment* parameter is a line of text that helps you identify and remember what your batch file does.

The only separators allowed in the *comment* are spaces, tabs, and commas.

In your batch file, you can use REM without a comment to add spacing for readability.

Example

The following example shows a batch file that uses remarks for both explanation and spacing:

```
REM This file formats and checks new disks
REM It is named CHECKNEW.BAT
REM
PAUSE Insert new disk in drive B
FORMAT B: /V
CHKDSK B:
```

SHIFT

Purpose

Lets you change the position of replaceable parameters in batch file processing.

Syntax

SHIFT

Comments

You can use the SHIFT command to change the positions of (replaceable) command line parameters.

Usually command files are limited to handling ten parameters, %0 through %9. But by using SHIFT, you can access more than ten parameters. This means that if there are more than ten parameters given on a command line, those that appear after the tenth (%9) will be shifted one at a time into %9.

You can use the SHIFT command even if you have less than ten parameters.

Note that there is no backward shift command. Once you have executed SHIFT, you cannot recover the first parameter (%0) that existed before the shift.

Example

The following file MYCOPY.BAT shows how to use the SHIFT command with any number of parameters. It copies a list of files to a specific directory.

```
REM MYCOPY.BAT copies
REM any number of files
REM to a directory.
REM The command is
REM mycopy dir files
:one
IF “%1” = “ ” GOTO two
SET todir = %1
SHIFT
COPY %1 %todir%
GOTO one
:two
SET todir=
ECHO All done
ECHO All done
```

Section 6

MS–DOS Editing and Function Keys

In this section you will learn about:

- the MS–DOS editing and function keys
- the editing template
- the MS–DOS control characters.

SPECIAL MS–DOS EDITING KEYS

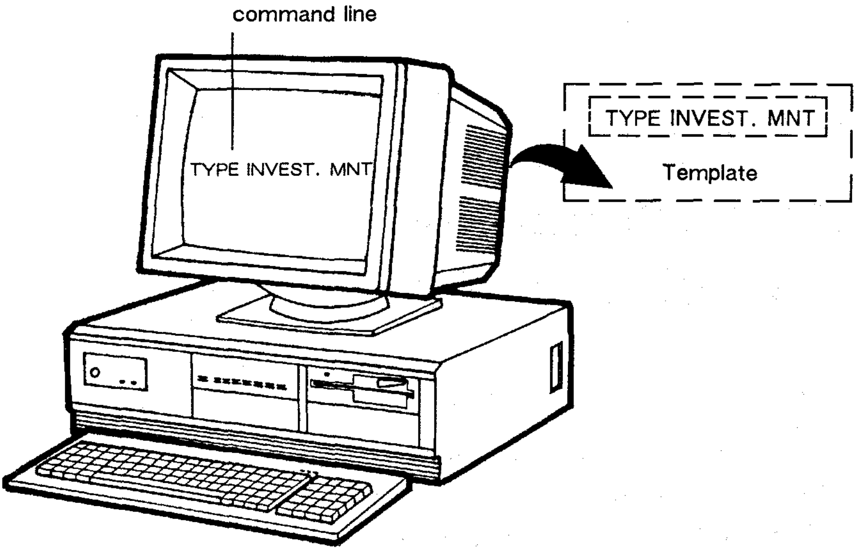
Many operating systems handle command input differently than MS–DOS does. One difference in particular that sets MS–DOS apart is its set of special editing keys. For instance, with MS–DOS you don't have to type the same sequences of keys repeatedly, because the most recently typed command line is automatically placed in a special storage area called a *template*.

By using the template and the special editing keys, you can take advantage of the following MS–DOS features:

- You can repeat a command instantly by pressing two keys.
- If you make a mistake in a command line, you can edit and retry it without having to retype the entire line.
- With a minimum of typing, you can edit and execute a command line that is similar to a previous one.

HOW MS–DOS USES THE TEMPLATE

When you type a command and press the Enter key, MS–DOS automatically sends it to the command processor (COMMAND.COM) for execution. At the same time, MS–DOS also sends a copy of this command to the template. You can then recall or modify the command by using the MS–DOS special editing keys. The following illustration shows how the template relates to the command line:



Special Editing Functions

KEY	EDITING FUNCTION
F1	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	Voids the current input and leaves the template unchanged.
Ins	Enters/exits insert mode.
F5	Makes the new line the new template.
F6	Puts a Ctrl-Z (1AH) end-of-file character in the new template.

Examples

Suppose you want to see the directory information for a file named INVEST.MNT. To get this information you could type the following command:

```
    DIR INVEST.MNT
```

This command line (DIR INVEST.MNT) is also saved in the template. If you want to repeat the command, just press two keys: **F3** and **Enter**.

MS–DOS displays the repeated command on the screen when you press F3 as shown below:

```
    DIR INVEST.MNT
```

Notice that when you press the F3 key, MS–DOS copies the contents of the template to the command line; pressing the Enter key then sends the command line to the command processor for execution.

If you want to display information about a file named INVEST.RPT, you can use the contents of the template. Pressing **F2** followed by the letter **m** copies all characters from the template to the command line, up to but not including the **m**. MS–DOS displays:

```
    DIR INVEST._
```

Note that the underline is your cursor. Now type the letters RPT to get the following result.

```
    DIR INVEST.RPT_
```

The command line (DIR INVEST.RPT) is now in the template and ready to be sent to the command processor for execution. To run the command, press the **Enter** key.

Now, assume that you want to run the following command:

```
    TYPE INVEST.RPT
```

To do this, type the word **TYPE** and then press the following sequence of keys: **Ins, space bar, F3, Enter.**

As you type, the characters appear directly on the command line, overwriting their corresponding characters in the template. Before you press the Ins key, the word TYPE replaces the word DIR (and the space following it) in the template. After you press the Ins key, this automatic replacement feature is turned off.

To insert a space between the word TYPE and the filename INVEST.RPT, you pressed Ins and then the space bar. Finally, to copy the rest of the template to the command line, you pressed F3 and then the Enter key. The command line **TYPE INVEST.RPT** has been processed by MS–DOS, and the template now looks like this:

```
TYPE INVEST.RPT.
```

If you had misspelled TYPE as PYTE, for example, a command error would have occurred. Still, instead of throwing away the whole command, you could save the misspelled line before pressing the Enter key. You can do this by pressing the **F5** key before the Enter key, creating a new template:

```
PYTE INVEST.RPT
```

You can then edit this error by pressing the following two keys: **F1** and **F3**. The F1 key copies a single character from the template to the command line, resulting in the command that you want:

```
TYPE INVEST.RPT
```

As an alternative, you can use the template that contains:

```
PYTE INVEST.RPT
```

Then you can use the Del and Ins keys to get the same result, as follows:

```
Del Del F1 Ins YP F3
```

To illustrate how the keys you type affect the command line, compare the key pressed with its result (shown beneath it along with a description of the effect).

```
Del
_                             Skips over 1st template character
Del
Del
_                             Skips over 2nd template character
F1
T_
                              Copies 3rd template character
Ins YP
TYP_
                              Inserts two characters, Y and P
F3
TYPE INVEST.RPT_
                              Copies rest of template
```

Notice that Del does not affect the command line. Instead, it affects the template by deleting the first character. Similarly, F4 deletes characters in the template, up to, but not including, a given character.

These special editing keys do give you more power and flexibility when you are typing. But, in addition to these keys, MS-DOS also has control characters that help you control the output from a command, or control the contents of the current command line. The next section describes how to use the MS-DOS control characters.

CONTROL CHARACTERS

A control character affects the command line in a special way. For example, you use Ctrl-C to stop running the current command, and you use Ctrl-S to suspend the screen output from a command.

Note that when you type a control sequence, such as Ctrl-C, you must hold down the **Ctrl** key and then press the **C** key.

Table 6-1 lists the MS-DOS control characters and describes what they do.

Table 6-1 Control Characters

CONTROL CHARACTER	WHAT IT DOES
Ctrl-C	Aborts the current command.
Ctrl-H	Removes the last character from a command line, and erases that character from the terminal screen.
Ctrl-J	Inserts a physical end-of-line, but does not empty the command line. Use the LINEFEED key to extend the current logical line beyond the physical limits of the terminal screen.
Ctrl-N	Causes echoing of output to a lineprinter.
Ctrl-P	Causes terminal output to a lineprinter.
Ctrl-S	Suspends output display on the screen. Press Ctrl-S again to resume.
Ctrl-X	Cancels the current line, empties the command line, and then outputs a backslash (\), Return, and LINEFEED. Ctrl-X does not affect the template used by the special editing commands.

Section 7

The Line Editor (EDLIN)

In this section you will learn:

- how to start EDLIN, the line editor program
- how to quit EDLIN
- how to use the MS–DOS special editing keys with EDLIN.

For information on specific EDLIN commands, see Section 8, “EDLIN Commands.”

ABOUT EDLIN

You can use the MS–DOS line editor, EDLIN, to create text files and save them on your disks, or to update existing files, saving both the original and the updated files. Or, with EDLIN you can delete, edit, insert, and display lines in files. It will also help you search for, and delete or replace, text within your files. And, though it isn’t a word processor, EDLIN does make it easy for you to create and revise files such as memos, letters, reports, or GW–BASIC programs.

How EDLIN Works

EDLIN divides the text from a file into lines, each line containing up to 253 characters. It gives each line a number and always numbers the lines consecutively. But, even though you see these line numbers on the screen when you use EDLIN, they are not part of the file.

When you insert lines of text in a file, the line numbers after the inserted text are automatically adjusted. Similarly, when you delete lines in a file, the line numbers following the deleted text are automatically renumbered.

How to Start EDLIN

To start EDLIN, you simply type the word EDLIN followed by a filename. If you are creating a new file, filename should be the name or pathname of the file you wish to create. If EDLIN does not find this file on the default disk drive, it creates a new file with the name or pathname that you specify. For example, if you want to create a file called budget.jun, you would type the following command and then press the Enter key:

```
EDLIN BUDGET.JUN
```

EDLIN would then display the following:

```
New file
*_
```

Note that the EDLIN prompt is an asterisk (*).

To begin entering text you must type an I (INSERT) command to insert lines. The I command is discussed later in this section. For now you can type lines of text into your file, or use any of the EDLIN commands. These are discussed in more detail in Section 8, "EDLIN Commands."

Be sure to press the Enter key at the end of each line.

Suppose you want to edit an existing file called BUDGET.MAY. To do this you would type the following:

```
EDLIN BUDGET.MAY
```

Then, when EDLIN finds the BUDGET.MAY file, it loads it into memory. If your computer has enough memory to load the entire file, EDLIN displays the following message:

```
End of input file
*
```

You can then edit the file by using EDLIN commands.

If the file is too large to be loaded into memory, EDLIN loads lines from the file until memory is 3/4 full, and displays the asterisk (*) prompt. You can then edit the portion of the file that is in memory.

To edit the rest of the file, you must save some of the edited lines on a disk to free memory. EDLIN will then be able to load the remaining unedited lines from the disk into memory. Refer to the W (WRITE) and A (APPEND) commands in Section 8, "EDLIN Commands," for instructions on editing large files.

How to Quit EDLIN

When you finish your editing session, you can save your original file and the updated (new) file by using the E (END) command, discussed in Section 8, "EDLIN Commands." EDLIN renames your original file with the extension .BAK, and saves the updated file with the filename and extension you gave when you started EDLIN.

CAUTION

> You cannot update a file with an extension of .BAK because when you try to save your file, EDLIN will always save the original file as .BAK, thereby losing your changes. If you need to edit such a file, rename it with another extension (using the MS-DOS REN command discussed in Section 4, "MS-DOS Commands"), and start EDLIN by using the new filename.

Special Editing Keys

To edit your text files you can also use the special editing keys and template introduced in Section 6, "MS-DOS Editing and Function Keys."

Table 7-1 summarizes the commands, codes, and functions of the special editing keys. Descriptions of these special editing keys follow the table:

Table 7-1 EDLIN Special Editing Keys

KEY	WHAT IT DOES
F1	Copies one character from the template to the new line.
F2	Copies all characters from the template to the new line, up to the character specified.
F3	Copies all remaining characters in the template to the screen.
Del	Does not copy (skips over) a character.
F4	Does not copy (skips over) the characters in the template, up to the character specified.
Esc	Clears the current input and leaves the template unchanged.
Ins	Enters/exits insert mode.
F5	Makes the new line the new template.

F1

Purpose

Copies one character from the template to the current line.

Comment

When you press the F1 key, EDLIN copies one character from the template to the current line, and turns insert mode off.

Example

As an example of how to use the F1 key with EDLIN, type the following line:

```
1:*Sharpe Office Supplies.
2:*_
```

At the beginning of the editing session, the cursor (shown by the underline) is at the beginning of the line. When you press the F1 key, EDLIN copies the first character S to line 2 as shown here:

```
F1
2:*S_
```

Each time you press the F1 key one more character appears:

```
F1
2:*Sh_

F1
2:*Sha_

F1
2:*Shar_
```

F2

Purpose

Copies multiple characters up to a given character.

Comments

When you press the F2 key, EDLIN copies all the characters, up to a given character, from the template to the current line. The given character is the one that you type immediately after F2. EDLIN does not copy or display the given character on the screen, but it does copy and display the characters from the template up to the position of that character. Of course, if the template does not contain the character, EDLIN does not copy anything.

If you use the F2 key, you automatically turn off insert mode.

Example

As an example of how to use the F2 key with EDLIN, type the following line:

```
1:*Sharpe Office Supplies.
2:*_
```

Remember that at the beginning of the editing session, the cursor (shown by the underline) is at the beginning of the line. When you press the F2 key followed by the letter c, EDLIN copies the characters up to the c in the word Office.

```
F2 c
2:*Sharpe Offi_
```

F3

Purpose

Copies the template to the current line.

Comments

When you press the F3 key, EDLIN copies the remaining characters in the template to the current line. No matter where the cursor is when you press the F3 key, EDLIN displays the rest of the line and leaves the cursor at the end of the line.

Example

As an example of how to use the F3 key with EDLIN, type the following line:

```
1:*Sharpe Office Supplies.
2:*_
```

Remember that at the beginning of the editing session, the cursor (shown by the underline) is at the beginning of the line. When you press the F3 key, EDLIN copies the characters in the template (shown in line 1) to the line with the cursor (shown in line 2):

```
F3
2:*Sharpe Office Supplies._
```

Also, this command automatically turns off insert mode.

Del

Purpose

Skips over one character in the template.

Comments

Each time you press the Del key, EDLIN skips over and does not copy the next character in the template. The action of the Del key is similar to the F1 key, except that Del skips a character in the template instead of copying it to the current line.

Example

As an example of how to use the Del key with EDLIN, type the following line:

```
1:*Sharpe Office Supplies.
2:*_
```

Remember that at the beginning of the editing session, the cursor (shown by the underline) is at the beginning of the line. When you press the Del key, EDLIN skips over the first character, S:

```
Del
2:*_
```

The cursor does not move as EDLIN changes the template. To see how much of the line has been skipped over, press the F3 key. This action moves the cursor past the last character of the line:

```
F3
2:*harpe Office Supplies._
```

F4

Purpose

Skips multiple characters in the template up to the specified character.

Comments

When you press the F4 key, EDLIN skips over all characters up to a given character in the template. EDLIN does not copy or display any of the characters up to and including the given character. Of course, if the template does not contain that character, EDLIN does not skip any characters.

Note that the action of the F4 key is similar to that of the F2 key, except F4 skips over characters in the template instead of copying them to the current line.

Example

As an example of how to use the F4 key with EDLIN, type the following line:

```
1:*Sharpe Office Supplies.
2:*_
```

Remember that at the beginning of the editing session, the cursor (shown by the underline) is at the beginning of the line. When you press the **F4** key followed by the letter **c**, EDLIN skips over all the characters in the template up to the *c* in the word *Office*:

```
F4 c
2:*_
```

The cursor does not move as EDLIN changes the template. To see how much of the line has been skipped over, press the **F3** key to copy the template. This action displays the rest of the line and moves the cursor to the end of the line:

```
F3
2:*ce Supplies_.
```

Esc

Purpose

Quits input and clears the current line.

Comments

When you press the **Esc** key, EDLIN empties the current line and leaves the template unchanged. Esc also prints a backslash (\), Return, and linefeed, and turns insert mode off. The cursor (shown by the underline) is at the beginning of the line. If you then press the **F3** key, EDLIN copies the template to the current line, making it appear as it was before you pressed the Esc key.

Example

As an example of how to use the Esc key with EDLIN, type the following lines:

```
1:Sharpe Office Supplies.
2:*The World Leader_
```

To cancel the current line, line 2, press **Esc**. Notice that a backslash appears on line 2 to tell you it has been canceled:

```
Esc
2:*The World Leader\
```

Press the **Enter** key to keep line 1, or to perform any other editing functions. Now if you press **F3**, EDLIN copies the original template to the line:

```
F3
2:Sharpe Office Supplies.
```

Ins

Purpose

Enters insert mode or replace mode.

Comments

When you start EDLIN, you are automatically in replace mode. The first time you press the Ins key, EDLIN enters insert mode. In insert mode the cursor in the template does not move, but in the current line it moves as you insert each character. When you finish inserting characters and press the Ins key again, EDLIN reenters replace mode with the cursor at the same character in the template as when you entered insert mode.

In insert mode, EDLIN puts characters that you type at the keyboard into the template and in front of the cursor on the current line.

In replace mode, EDLIN replaces characters in the template and on the current line with characters that you type at the keyboard.

Examples

As an example of how to use the Ins key with EDLIN, type the following line:

```
1:*Sharpe Office Supplies.
2:*_
```

Remember that at the beginning of the editing session, the cursor (shown by the underline) is at the beginning of the line. First, press the **F2** and **O** keys:

```
F2 O
2:*Sharpe_
```

Second, press the **Ins** key and type the word **Automatic** followed by a space:

```
Ins Automatic
2:*Sharpe Automatic_
```

If you now press the F3 key, EDLIN copies the rest of the template to the line:

```
F3
2:*Sharpe Automatic Office Supplies_.
```

If you press the Enter key after entering insert mode, EDLIN does not copy the remainder of the template and ends the current line after the inserted text:

```
1: Sharpe Office Supplies.
2:*Sharpe Automatic
3:*_
```

To exit insert mode and enter replace mode, simply press the **Ins** key again. Now all the characters you type will replace characters in the template.

For a further example of how to use the Ins key, type the following line:

```
1:*Sharpe Office Supplies.
2:*_
```

Once again, remember that at the beginning of the editing session, the cursor (shown by the underline) is at the beginning of the line. Now type the following sequence of keys and words (the results are shown below each key sequence):

F2 c

```
2:*Sharpe Offi_
```

Ins cial

```
2:*Sharpe Official_
```

Ins Sharpeware.

```
2:*Sharpe Official Sharpeware._
```

Notice that you inserted **cial** and replaced **ce Supplies** with **Sharpeware.**

The template now contains **Sharpe Official Sharpeware.**

F5

Purpose

Creates a new template.

Comments

When you press the **F5** key, EDLIN copies the current line to the template and deletes the previous contents. Pressing **F5** also displays an @ ("at" symbol), and outputs a Return and a linefeed. When you press **F5**, EDLIN empties the current line and turns off insert mode.

F5 performs the same function as the Esc key, except that it changes the template, printing an @ instead of a backslash.

Example

As an example of how to use the F5 key with EDLIN, type the following line:

```
1:*Sharpe Office Supplies.
2:*_
```

Remember that at the beginning of the editing session, the cursor (shown by the underline) is at the beginning of the line. Now type the following sequence of keys and words (the results are shown below each key sequence):

F2 c

```
2:*Sharpe Offi_
```

Ins cial

```
2:*Sharpe Official_
```

Ins Sharpeware.

```
2:*Sharpe Official Sharpeware._
```

At this point, suppose you want to add a word at the beginning of this line, but you don't want to backspace and retype the whole line. Just press the **F5** key to put the current line into the template:

```
1: Sharpe Office Supplies.
2:*Sharpe Official Sharpeware._
```

The @ shows that this new line is now the new template. To add the word **Introducing:**, followed by a space, at the beginning of the line, press **Ins** and type the following sequence of keys and words (the results are shown below each key sequence):

Ins Introducing:

```
2:*Introducing:_
```

Then press the **F3** key to insert the contents of the template.

F3

```
2:*Introducing: Sharpe Official Sharpeware._
```

Section 8

EDLIN Commands

This section describes in detail the EDLIN commands, listed in Table 8-1.

Table 8-1 EDLIN commands

COMMAND	NAME	WHAT IT DOES
A	Append	Appends lines.
C	Copy	Copies lines.
D	Delete	Deletes lines.
line	Edit	Edits a line or lines.
E	End	Ends editing.
I	Insert	Inserts lines of text.
L	List	Lists a range of lines.
M	Move	Moves a range of text to a specified line.
P	Page	Pages through a file 23 lines at a time.
Q	Quit	Quits the editing session without saving the file.
R	Replace	Replaces text.
S	Search	Searches for text.
T	Transfer	Transfers the contents of another file into the file being edited.
W	Write	Writes specified lines to disk.

SOME TIPS FOR USING EDLIN COMMANDS

Once you have started editing a file with EDLIN, you can use the EDLIN commands to edit lines of text in it. Here are a few things to remember when using EDLIN commands:

- You can use pathnames in commands. For example, by typing the following command you can edit a file named REPORT.MAY in a subdirectory named \SHARPE\BUDGET:

  ```
  EDLIN \SHARPE\BUDGET\REPORT.MAY
  ```

- You can reference line numbers relative to the current line, which EDLIN shows with an asterisk (*). To indicate lines before the current line, use a minus sign with a number; to indicate lines after the current line, use a plus sign with a number. The following command, for example, lists 10 lines before the current line, the current line, and 10 lines after the current line :

  ```
  -10,+10L
  ```

- You can type EDLIN commands with or without a space between the line number and command. For example, for deleting line 6, the command 6D is the same as 6 D.

- You can type multiple commands on one command line. Just type them one after another. But if you want to use the EDLIN line (edit) command to edit a specific line, you have to separate the line number from the other commands with a semicolon.

  To end a string in an S (SEARCH) or R (REPLACE) command, just press **Ctrl-Z** instead of the Enter key. The following command line, for example, edits line 15, then displays lines 10 through 20 on the screen:

  ```
  15;-5,+5L
  ```

  The command line in the next example searches for the words "monthly budget" and then displays five lines before and five lines after the line that contains "monthly budget". Also, in this example and in those that follow, you'll see control key sequences. You don't really type "Ctrl"; instead, you simply press the **Ctrl** key followed by the control character, **Z** or **C** or **V**.

  ```
  S monthly budget Ctrl-Z -5,+5L
  ```

  If EDLIN can find any lines that contain the words "monthly budget", then the displayed lines are the five lines before and the five lines after the current line. Note that the current line (the line with the asterisk) must be *before* the line or lines that contain the search string.

- You can insert a control character, such as Ctrl–C, into text by using the quote character, Ctrl–V, before it while in insert mode. Ctrl–V tells MS–DOS to recognize the next capital letter typed as a control character. You can also put a control character in an S (SEARCH) or R (REPLACE) command by using the quote character. For example, the following command finds the first occurrence of Ctrl–Z in a file:

  ```
  S Ctrl–V Z
  ```

  The next example replaces all occurrences of Ctrl–Z in a file with "pencil".

  ```
  R Ctrl–V Z Ctrl–Z pencil
  ```

  And this next command replaces all occurrences of Ctrl–C with "pen".

  ```
  s Ctrl–V C Ctrl–Z pen
  ```

  It is also possible to insert Ctrl–V into the text by typing **Ctrl–V V**.

- The Ctrl–Z character ordinarily tells EDLIN, "This is the end of the file." If you have Ctrl–Z characters elsewhere in your file, you must tell EDLIN that these other control characters do not mean end–of–file. To tell EDLIN to ignore the Ctrl–Z characters in the file and to show you the entire file, use the /B switch. For example, when you start EDLIN and want to ignore all Ctrl–Z characters in a file, you could use the /B option with the EDLIN command and your filename.

EDLIN COMMAND OPTIONS

Many EDLIN commands accept one or more options. The effect of a command option varies, depending on which command you use it with. The following list describes each option.

The Line Option

The *line* option is a line number that you type. Use a comma or space to separate the numbers from other line numbers, other options, and from the command. You can specify *line* in one of three ways:

number

Any number less than 65534. If you specify a number larger than the largest existing line number, then *line* refers to the line after the last line number.

(period)

If you specify a period for *line*, it refers to the current line number. The current line is the last line you edited, not necessarily the last line you displayed. EDLIN marks the current line with an asterisk (*) between the line number and the first character.

# (pound sign)

The pound sign indicates the line after the last line number. If you type # for *line*, it is the same as typing the last line number plus one.

Enter key

If you type a command and then press the **Enter** key without any of the line markers in this list, EDLIN uses a default value for each command (default values may be different for each command).

The Question Mark Option

The question mark (?) option tells EDLIN to ask you if the correct string has been found. You use the question mark only with the R (replace) and S (search) commands. Before continuing, EDLIN waits for you to type a **Y** or press **Enter** for a "Yes" response, or to press any other key for a "No" response.

The Text Option

The *text* option specifies text to be found, to be replaced, or to replace other text. Use the text option only with the S (search) and R (replace) commands. You must end each string of text with a Ctrl–Z or a Return (see the R command for details). You shouldn't leave any spaces between strings of text or between a string of text and its command letter, unless you want those spaces to be part of the text.

The remaining pages in this section describe the EDLIN commands. Each description explains the purpose and correct usage (syntax) of the command. Also, each command has several comments and examples, which offer advice, help, and even some shortcuts for using EDLIN.

APPEND

Purpose

Adds a specified number of lines from your disk to the file being edited in memory.

Syntax

[*n*]A

Comments

The *n* option specifies the number of lines that you want to read into memory. You need this command only if the file you want to edit is too large to fit into memory. When you start EDLIN, it reads as many lines as possible into memory.

To edit the remainder of the file that will not fit into memory, you must write lines that you have already edited onto your disk. Then you can load unedited lines from your disk into memory by using the A (APPEND) command. Refer to the W (WRITE) command in this section for information on how to write edited lines to your disk.

Note: if you do not specify the number of lines to append, EDLIN adds lines to the available memory until it is 3/4 full, but does nothing if available memory is already 3/4 full.

After the A command reads the last line of the file into memory, EDLIN displays the message "End of input file."

COPY

Purpose

Copies a range of lines to a specified line number, and when used with the *count* option, copies this range as many times as you want.

Syntax

[*line*],[*line*],*line*[,*count*]c

Comments

The first and second *line* options specify the range of lines that you want to copy. If you omit the first or second *line* option, EDLIN defaults to the current line. The third *line* option specifies the line before which EDLIN will place the copied lines.

You must not overlap the line numbers or you will get an "Entry error" message. For example, the following command would result in an error message:

```
3,20,15C
```

If you do not specify a number for the *count* option, EDLIN copies the lines one time and automatically renumbers the file after the copy.

Examples

As an example of how to use the C (COPY) command with EDLIN, type the following lines:

```
1: Sharpe Office Supplies
2: The World Leader in Office Sharpeware
3: I.R. Sharpe, President
4:
5: "You oughta be Sharpe too."
```

You could copy this entire block of text by typing the following command:

```
1,5,6C
```

This command copies lines 1 through 5 and duplicates them one time, beginning on line 6. The result is:

```
1: Sharpe Office Supplies
2: The World Leader in Office Sharpeware
3: I.R. Sharpe, President
4:
5: "You oughta be Sharpe too."
6: Sharpe Office Supplies
7: The World Leader in Office Sharpeware
8: I.R. Sharpe, President
9:
10: "You oughta be Sharpe too."
```

Now if you want to place this text within other text, you would specify the third *line* option as the line you want the copied text to appear before.

For example, suppose you want to copy lines and insert them in the middle of the file. The command **7,10,5C** would result in the following:

```
1: Sharpe Office Supplies
2: The World Leader in Office Sharpeware
3: I.R. Sharpe, President
4:
5: The World Leader in Office Sharpeware
6: I.R. Sharpe, President
7:
8: "You oughta be Sharpe too."
9: "You oughta be Sharpe too."
10: Sharpe Office Supplies
11: The World Leader in Office Sharpeware
12: I.R. Sharpe, President
13:
14: "You oughta be Sharpe too."
```

DELETE

Purpose

Deletes a specified range of lines in a file.

Syntax

[*line*][,*line*]D

Comments

If you omit the first *line* option, EDLIN defaults to the current line (the line with the asterisk next to the line number). If you omit the second *line* option, then EDLIN deletes just the first line. Remember, too, that when you delete lines, EDLIN automatically renumbers the file.

Examples

Suppose that the following file exists and is ready to edit:

```
 1: Dear Mr. Dimm,
 2:
 3: I was sorry to hear of your recent
 4: hospitalization due to electrical
 5: shock from our X-1000 Automatic
 6: Pencil Sharpener.
 7:
 8: As a result of your accident, we
 9: are redesigning our manual
10: to warn our customers against trying
11: to sharpen metal objects.
12:
13: We hope that you will be more
14: careful with electrical appliances
15: in the future.
16:
17: Sincerely,
18:
19:*I.R. Sharpe, President
```

But now, say you decide not to warn Mr. Dimm about being careful. To delete lines 12 through 15, type the following

```
12,15D
```

The result of this command is:

```
 1: Dear Mr. Dimm,
 2:
 3: I was sorry to hear of your recent
 4: hospitalization due to electrical
 5: shock from our X-1000 Automatic
 6: Pencil Sharpener.
 7:
 8: As a result of your accident, we
 9: are redesigning our manual
10: to warn our customers against trying
11: to sharpen metal objects.
12:
13: Sincerely,
14:
15:*I.R. Sharpe, President
```

To close up the space you might decide to delete line 7. You could just type the command **7D**. The result would be:

```
 1: Dear Mr. Dimm,
 2:
 3: I was sorry to hear of your recent
 4: hospitalization due to electrical
 5: shock from our X-1000 Automatic
 6: Pencil Sharpener.
 7:*As a result of your accident, we
 8: are redesigning our manual
 9: to warn our customers against trying
10: to sharpen metal objects.
11:
12: Sincerely,
13:
14: I.R. Sharpe, President
```

Suppose, though, that you just want a quick message that doesn't take responsibility for the accident. To delete a range of lines beginning with the current line, line 7, through line 11, type the following:

```
,11D
```

The result of this command is:

```
 1: Dear Mr. Dimm,
 2:
 3: I was sorry to hear of your recent
 4: hospitalization due to electrical
 5: shock from our X-1000 Automatic
 6: Pencil Sharpener.
 7:
 8: Sincerely,
 9:
10: I.R. Sharpe, President
```

Notice that as you delete lines, EDLIN automatically renumbers the remaining lines in the file.

EDIT

Purpose

Edits a line of text.

Syntax

[*line*]

Comments

The *line* option specifies the line of text you want to edit. When you type a line number as a command, EDLIN displays the line number and the text on that line; then, on the line below, EDLIN reprints the line number. Now you can retype the line, or use the EDLIN editing keys to edit it. The existing text of the line serves as the template until you press the **Enter** key.

If you do not type a line number (that is, if you only press the **Enter** key), EDLIN edits the line after the current line. (The current line is shown by an asterisk (*)).

If you do not need to change the current line and if the cursor is at the beginning or end of the line, you can simply press the **Enter** key to accept the line.

Note: if you press the **Enter** key while the cursor is in the middle of a line, EDLIN deletes the remainder of the line.

Example

Suppose that the following file exists and is ready to edit:

```
1: Dear Mr. Dimm,
2:
3: I was sorry to hear of your recent
4: hospitalization due to electrical
5: shock from our Automatic
6: Pencil Sharpener.
```

In line 5, say you want to insert the product's name, X-1000. To edit line 5, type the number 5. EDLIN then displays the contents of the line with the cursor below the line:

```
5:*shock from our Automatic
5:*_
```

Now you simply use the F2 key to skip to the A in the word Automatic, and type:

```
F2 A Ins X–1000
5:*shock from our X–1000

F3 Enter
 5:*shock from our X–1000 Automatic
```

At the EDLIN prompt, type L to see the file:

```
1: Dear Mr. Dimm,
2:
3: I was sorry to hear of your recent
4: hospitalization due to electrical
5:*shock from our X–1000 Automatic
6: Pencil Sharpener.
```

END

Purpose

Ends the editing session.

Syntax

E

Comments

The E (END) command saves the edited file on your disk, renames the original input file *filename*.BAK, and then exits EDLIN. If you created the file during the current editing session, EDLIN does not create a backup (.BAK) file.

The E command takes no options. This means that you must select the drive that you want to save the file on when you start EDLIN; if you don't, EDLIN saves the file on the disk in the default drive. However, you can still copy the file to a different drive by using the MS-DOS COPY command.

Before using the E command to save your file, make sure that the disk contains enough free space for the entire file. If it doesn't, EDLIN may not be able to write the entire file to the disk. The edited file will be lost, even though EDLIN may have saved part of it on the disk.

Example

To end your editing session, type **E** and then press **Enter**. After EDLIN processes the E command, the screen displays the MS-DOS prompt (for example, A>).

INSERT

Purpose

Inserts text immediately before the specified line.

Syntax

[*line*]I

Comments

If you are creating a new file, you must type the I (INSERT) command before you can insert a new line of text. Text begins on line 1, and each time you press the Enter key the next line number appears automatically.

EDLIN remains in insert mode until you press Ctrl–C. When you finish the insertion and exit insert mode, the line immediately following the inserted lines becomes the current line. EDLIN automatically increments the line numbers that follow the inserted section by the number of lines that you inserted.

If you do not specify *line*, the default is the current line number and EDLIN inserts the lines before the current line. If line is a number larger than the last line number, or if you specify a pound sign (#) as *line*, EDLIN appends the inserted lines to the end of the file. In this case, the last line that you inserted becomes the current line.

Examples

Suppose the following file exists and is ready to edit:

```
 1: Dear Mr. Dimm,
 2:
 3: I was sorry to hear of your recent
 4: hospitalization due to electrical
 5: shock from our X–1000 Automatic
 6: Pencil Sharpener.
 7:
 8: Sincerely,
 9:
10: I.R. Sharpe, President
```

Since this letter doesn't really offer any compensation for the accident, you might decide to add a brief note about a small gift that you're enclosing for Mr. Dimm. To insert text before line 8, type **8I**. The result is:

```
8:*_
```

Now type the following lines, which will begin on line 7:

```
 8:*As a result of your accident, we
```

Press the **Enter** key at the end of each line, and continue typing the next line:

```
 9:*are redesigning our manual to
10:*warn our customers against trying
11:*to sharpen metal objects.
```

To end the insertion, press Ctrl–C on the next line, and then type L to list the file. The result is:

```
 1: Dear Mr. Dimm,
 2:
 3: I was sorry to hear of your recent
 4: hospitalization due to electrical
 5: shock from our X–1000 Automatic
 6: Pencil Sharpener.
 7:
 8: As a result of your accident, we
 9: are redesigning our manual to
10: warn our customers against trying
11: to sharpen metal objects.
12:*Sincerely,
13:
14: I.R. Sharpe, President
```

To insert a blank line immediately before the current line (line 12), type I. The result is:

```
12:*_
```

Insert a blank line by pressing **Enter,** and end the insertion by pressing **Ctrl–C** on the next line. Then, to list the file and see the result, type **L**.

```
 1: Dear Mr. Dimm,
 2:
 3: I was sorry to hear of your recent
 4: hospitalization due to electrical
 5: shock from our X-1000 Automatic
 6: Pencil Sharpener.
 7:
 8: As a result of your accident, we
 9: are redesigning our manual to
10: warn our customers against trying
11: to sharpen metal objects.
12:
13:*Sincerely,
14:
15: I.R. Sharpe, President
```

To append new lines to the end of the file, enter **16I**. This produces the following:

```
16:*_
```

Now type the following new lines:

```
16: Sharpe Office Supplies
17: The World Leader in Office Sharpeware
18: Our motto: "You oughta be Sharpe too"
```

To get out of insert mode, press **Ctrl-C** on line 19. The new lines will appear at the end of all previous lines in the file. Now list all the lines by typing **1,23L**. The result is:

```
 1: Dear Mr. Dimm,
 2:
 3: I was sorry to hear of your recent
 4: hospitalization due to electrical
 5: shock from our X-1000 Automatic
 6: Pencil Sharpener.
 7:
 8: As a result of your accident, we
 9: are redesigning our manual to
10: warn our customers against trying
11: to sharpen metal objects.
12:
13: Sincerely,
14:
15: I.R. Sharpe, President
16: Sharpe Office Supplies
17: The World Leader in Office Sharpeware
18: Our motto: "You oughta be Sharpe too"
```

LIST

Purpose

Lists a range of lines, including the two lines specified.

Syntax

[*line*] [,*line*] L

Comments

EDLIN provides default values if you do not type either option. If you do not type the first *line* option, as in the following command, the display will start 11 lines before the current line and end with the specified *line*:

> ,*line* L

The beginning comma is needed to show that you omitted the first *line* option.

If the specified *line* is more than 11 lines before the current line, the display will be the same as if you omitted both options.

EDLIN displays 23 lines, starting with the specified *line*, if you omit the second option, as in the following command:

> *line* L

If you just type L, EDLIN displays 23 lines--the 11 lines before the current line, the current line, and the 11 lines after the current line. If there are less than 11 lines before the current line, EDLIN displays more than 11 lines after the current line, up to a total of 23 lines.

Examples

Suppose the following file exists and is ready to edit:

```
 1: Dear Mr. Dimm,
 2:
 3: I was sorry to hear of your recent
 4: hospitalization due to electrical
 5: shock from our X-1000 Automatic
 6: Pencil Sharpener.
  .
  .
  .
15: We also intend to attach a metal
16: detector to future models.
  .
  .
  .
26: I.R. Sharpe, President
27: Sharpe Office Supplies
28: The World Leader in Office Sharpeware
29: Our motto: "You oughta be Sharpe too"
```

To list a range of lines without referring to the current line, type **2,6L**. The result is:

```
 2:
 3: I was sorry to hear of your recent
 4: hospitalization due to electrical
 5: shock from our X-1000 Automatic
 6: Pencil Sharpener.
```

To list a range of lines beginning with the current line, enter **16,L** or **,L**. The result is:

```
16:*detector to future models.
  .
  .
  .
26: I.M. Sharpe, President
27: Sharpe Office Supplies
28: The World Leader in Office Sharpeware
29: Our motto: "You oughta be Sharpe too"
```

To list a range of 23 lines centered around the current line, enter **L**. The result is:

```
 5: shock from our X-1000 Automatic
 6: Pencil Sharpener.
 .
 .
 .
15: We also intend to attach a metal
16:*detector to future models.
 .
 .
 .
26: I.R. Sharpe, President
27: Sharpe Office Supplies
```

MOVE

Purpose

Moves a range of text to the specified *line*.

Syntax

[*line*],+*line*,*line*M

Comments

The M (MOVE) command lets you move a block of text to another location in a file. The first and second *line* options specify the range of lines that you want to move. The third *line* option specifies the line you want to move the first line in the range to.

EDLIN automatically renumbers the lines after it moves them.

For example, the following command moves the text from the current line--plus 25 lines--to line 100:

```
,+25,100M
```

If the line numbers that you specify overlap, EDLIN displays an "Entry error" message.

Example

Suppose the following file exists and is ready to edit.

```
 1: Dear Mr. Dimm,
 2:
 3: I was sorry to hear of your recent
 4: hospitalization due to electrical
 5: shock from our X-1000 Automatic
 6: Pencil Sharpener.
 7:
 8: As a result of your accident, we
 9: are redesigning our manual to
10: warn our customers against trying
11: to sharpen metal objects.
12:
13: Sincerely,
14:
15: I.R. Sharpe, President
16: Sharpe Office Supplies
17: The World Leader in Office Sharpeware
18: Our motto: "You oughta be Sharpe too"
```

What if you prefer to have the motto at the start of the letter? If so, you could move lines 16--18 to line 1 by typing the following command:

```
16,18,1M
```

The result of this command is:

```
 1: Sharpe Office Supplies
 2: The World Leader in Office Sharpeware
 3: Our motto: "You oughta be Sharpe too"
 4: Dear Mr. Dimm,
 5:
 6: I was sorry to hear of your recent
 7: hospitalization due to electrical
 8: shock from our X-1000 Automatic
 9: Pencil Sharpener.
10:
11: As a result of your accident, we
12: are redesigning our manual to
13: warn our customers against trying
14: to sharpen metal objects.
15:
16: Sincerely,
17:
18: I.R. Sharpe, President
```

PAGE

Purpose

Displays a file one page (23 lines) at a time.

Syntax

[*line*] [,*line*]P

Comments

The first *line* option specifies the line at which EDLIN starts displaying. The second *line* option specifies how many lines appear on each page. If you do not type the first *line*, EDLIN starts the page at the line after the current line. If you do not type the second line option, EDLIN lists 23 lines on each page.

QUIT

Purpose

Quits the editing session, does not save any editing changes, and exits to the MS–DOS operating system.

Syntax

Q

Comments

This command is useful if you don't want to make any changes to a file. If you use the Q (QUIT) command, EDLIN prompts you to make sure you don't want to save the changes. If you do want to save your changes, use the E (END) command.

If you want to quit the editing session, type **Y** (for Yes) when prompted; remember, though, that EDLIN will not save your editing changes and will not create a backup (.BAK) file. Refer to the E command in this section for information about the .BAK file.

If you want to continue the editing session, type **N** (for No).

CAUTION

When you exit EDLIN, it erases any previous copy of the file that has a .BAK extension. If you reply Y to the "Abort edit (Y/N)?" message, EDLIN deletes your previous backup copy.

Example

The following example shows how to quit EDLIN without saving your changes. First, to get the "Abort edit (Y/N)?" message, type Q. Then, when the message appears, type **Y** and press **Enter**.

REPLACE

Purpose

Replaces all occurrences of a string of text in a range with a different string of text.

Syntax

[*line*] [,*line*] [?]R*text1* **Ctrl–Z** *text2*

Comments

The first and second *line* options specify the range of lines that the R (REPLACE) command uses. Each time EDLIN finds *text1*, it replaces it with *text2*, displaying each line that changes. If a line contains two or more replacements, it is displayed once for each change. When EDLIN has made all the changes, the R command ends and the asterisk prompt reappears.

If you want to replace one string of text with another, you must separate the two with a Ctrl–Z. You can end the second string by pressing the **Enter** key.

If you do not specify *text1*, the R command assumes the old (the previous) value. If this is the first replacement that you have made during the current editing session, and if you do not specify *text1*, the command ends. If you do not specify *text2*, you must end text1 by pressing the **Enter** key.

If you omit the first *line* option, EDLIN uses the line after the current line, by default. The default for the second *line* option is #. Remember that # refers to the line after the last line of the file.

If you end *text1* with a Ctrl–Z and do not specify *text2*, EDLIN assumes you want blank spaces for *text2*. For example, suppose you want to delete all occurrences of the word "clients" from your file. To do this you could simply type the following command, then press **Ctrl–Z** and **Enter**:

```
    Rclients
```

The next command replaces "clients" with the previous *text2*:

```
    Rclients
```

The following command makes the previous *text1* become the previous *text2*:

```
    R
```

Note that "previous" refers to an earlier string of text specified in an S or R command.

If you specify a question mark (?) in the R command, EDLIN stops at each line with text that matches *text1*, displays the line with *text2*, and then displays the prompt "O.K?" If you press **Y** or **Enter**, *text2* replaces *text1*, and EDLIN looks for the next occurrence of *text1*. It continues this process it until reaches the end of the range or the end of the file. After EDLIN finds the last *text1*, it displays the asterisk prompt.

If you press any key other than **Y** or the **Enter** key after the "O.K.?" prompt, *text1* is left as is, and EDLIN moves to the next occurrence of *text1*. If *text1* occurs more than once in a line, EDLIN individually replaces each occurrence of *text1*, and displays the "O.K.?" prompt after each replacement. In this way you can replace only the desired text1 strings and prevent unwanted substitutions.

Examples

Suppose the following file exists and is ready for editing:

```
 1: Dear Mr. Dimm,
 2:
 3: I was sorry to hear of your recent
 4: hospitalization due to electrical
 5: shock from our X-1000 Automatic
 6: Pencil Sharpener.
 7:
 8: As a result of your accident, we
 9: are redesigning our manual to
10: warn our customers against trying
11: to sharpen metal objects.
12:
13: Sincerely,
14:
15: I.R. Sharpe, President
```

Now, suppose that in lines 5 through 10 you want to replace all occurrences of the word "our" with the word "the". To do this you would simply type **5,10 Rour**, press **Ctrl-Z**, type **the**, and press the **Enter** key. The result is:

```
 5: shock from the X-1000 Automatic
 8: As a result of ythe accident,
 9: are redesigning the manual to
10: warn the customers against
```

In the previous example, some unwanted changes occurred. To avoid these and to confirm each replacement, you can use the same file with a slightly different command.

In the next example you will see how to replace only certain occurrences of "our" with "the". At the EDLIN prompt, type the following sequence of keys and words, and then press the **Enter** key:

```
1,15? Rour Ctrl-Z the
```

The result is:

```
 5: shock from the X-1000 Automatic
O.K.? y
 8: As a result of ythe accident, we
O.K.? n
 9: are redesigning the manual to
O.K.? y
10: warn the customers against trying
O.K.? n
*
 _
```

Type the LIST command, **L**, to see the result of all these changes:

```
 5: shock from the X-1000 Automatic
 ..
 8: As a result of your accident, we
 9: are redesigning the manual to
10: warn our customers against trying
 .
 .
```

SEARCH

Purpose

Searches a range of lines for a string of text.

Syntax

[*line*] [,*line*] [?]S*text*

Comments

The first and second *line* options specify the range of lines for EDLIN to search. You end the text option by pressing the Enter key. EDLIN displays the first line that matches the string; that line then becomes the current line. Unless you type the question mark (?) option, the S (SEARCH) command ends when it finds the first match. If EDLIN cannot find a line with a match, it displays the message "Not found."

If you include the question mark option (?), EDLIN displays the first line with matching text and prompts you with the message "O.K.?". If you press either **Y** (Yes) or the **Enter** key, this line becomes the current line and the search ends. If you press any other key, the search continues until another match is found, or until all lines have been searched. (The search ends when EDLIN displays the "Not found" message.)

If you don't type the first line number, EDLIN defaults to the line *after* the current line; if you don't type the second line number, it defaults to # (the line after the last line of the file).

If you omit the *text* option, EDLIN uses the text from the previous S or R (replace) command. If this is the first S or R command you have used during the current session, and you have not specified a search string, the S command ends immediately.

Examples

Suppose the following file exists and is ready for editing:

```
 1: Dear Mr. Dimm,
 2:
 3: I was sorry to hear of your recent
 4: hospitalization due to electrical
 5: shock from our X-1000 Automatic
 6: Pencil Sharpener.
 7:
 8: As a result of your accident, we
 9: are redesigning our manual to
10: warn our customers against trying
11: to sharpen metal objects.
12:
13: Sincerely,
14:
15: I.R. Sharpe, President
```

To search for the first occurrence of the word to, type the command **2,12 Sto** and press the **Enter** key. EDLIN displays the following lines:

```
3: I was sorry to hear of your recent
```

To get the to in line 4, modify the S command by pressing **Del F3** followed by the **Enter** key. The search continues at the line after the current line (i.e., the search starts at line 4), because you deleted the first line number in the template. The result is:

```
4: hospitalization due to electrical
```

To search through several occurrences of a string until the correct string is found, type the command **1, ? Sto**.The result is:

```
3: I was sorry to hear of your recent O.K.?_
```

If you press any key (except **Y** or the **Enter** key), the search continues, so type **N** here:

```
O.K.? N
```

Continue:

```
4: hospitalization due to electrical O.K.?_
```

Now press **Y** to terminate the search:

```
O.K.?Y
*_
```

EDLIN reports a match and continues to search for the same string when you retype the **S** command and press **Enter:**

```
S
```

EDLIN reports another match, if there is one.

```
S
```

EDLIN displays the "Not found" message at the end of the search.

Note that the text string defaults to any string specified by a previous R or S command.

TRANSFER

Purpose

Inserts, at a specific line number, the contents of another file into the file that you are currently editing.

Syntax

[*line*]T*filename*

Comments

The T (TRANSFER) command puts the contents of one file into another file, or into the text you are typing. EDLIN inserts *filename* at the line number you give in the *line* option, and then automatically renumbers the lines. If you omit the line number, EDLIN inserts the text, beginning on the current line.

Example

To copy a file named IRSHARPE.MEM to line 12 of the file you are editing, use the following command:

```
12 T IRSHARPE.MEM
```

WRITE

Purpose

Writes a specific number of lines to a disk.

Syntax

[*n*]W

Comments

The *n* option specifies the number of lines that you want to write to the disk. You need this command only if the file you are editing is too large to fit into memory. When you start EDLIN, it reads lines from your file until memory is 3/4 full.

To edit the remainder of your file, you must write the edited lines in memory to your disk. Then you can load additional unedited lines from your disk into memory by using the A (APPEND) command, which is described earlier in this section.

If you do not specify the number of lines for EDLIN to write, it writes lines until memory is 3/4 full. But it does not write any lines to your disk until memory is more than 3/4 full. Also, EDLIN renumbers all of the lines so that the first remaining line becomes line number 1.

Section 9

File Comparison Utility (FC)

In this section you will learn:

- how to compare files line by line
- how to compare files byte by byte.

INTRODUCTION

It is sometimes useful to compare files on your disk. If you have copied a file and later want to compare copies to see which one is current, you can use the MS–DOS file comparison utility, FC.

The FC utility compares the contents of two files, or two sets of files. The listing of the differences between the two files can be displayed on the screen or written to a third file. The files being compared must be text files––unless you specify the /B switch.

The comparisons are made in one of two ways:

- line by line
- byte by byte.

The line–by–line comparison isolates blocks of lines that are different between two files and prints those blocks of lines. You use this type of comparison to check for differences between text files (also known as ASCII files). The byte–by–byte comparison displays the bytes that are different between two files. You use this type of comparison to check for differences between binary files.

LIMITATIONS ON COMPARISONS

FC uses a large amount of memory (enough to hold 100 lines) as buffer storage space to hold the text files. If these files are larger than available memory, FC will compare what it can load into the buffer space. If it doesn't find a match in those portions of the files in the buffer space, FC stops and displays the following message:

```
resynch failed. Files are too different.
```

For binary files larger than available memory, FC compares both files completely, overlaying the portion in memory with the next portion from disk. All differences are output in the same manner as for those files that fit completely in memory.

HOW TO USE FC

The syntax of FC is as follows:

For ASCII comparison –

FC [/A] [/C] [/L] [/LB *n*] [/N] [/T] [/W][/*nnnn*]
[*drive:*] *pathname1* [*drive:*] *pathname2*

For binary comparison –

FC [/B] [/*nnnnn*][*drive:*] *pathname1*[*drive:*]*pathname2*

The *pathname* option specifies the first file or set of files that you want to compare, while the *pathname* option specifies the second file or set of files that you want to compare. FC matches the first file against the second and reports any differences between them.

For example, to compare two text files called MONTHLY.REP and SALES.REP, you would type the following command:

```
FC /A MONTHLY.REP SALES.REP
```

If you want to specify a set of files, you simply use a wildcard as part of the filename. For example, if Pete and Betty both have files called REPORT1.TXT, REPORT2.TXT, and REPORT3.TXT, they might use FC to see if their files are identical. They would use the following command:

```
FC C:\USER\PETE\REPORT?.TXT \USER\BETTY\REPORT?.TXT
```

FC would take REPORT1.TXT in the \USER\PETE directory of disk drive C and compare it with the corresponding REPORT1.TXT in the \USER\BETTY directory. It would then compare Pete's REPORT2.TXT against Betty's, and, finally, compare their REPORT3.TXT files.

Since in this example no drive was specified for the second set of files, FC would have assumed that the \USER\BETTY directory was on the disk in the default drive.

FC SWITCHES

There are nine switches that you can use with the File Comparison utility. They are described in the following paragraphs.

The /A switch abbreviates the output of an ASCII comparison. Instead of displaying all of the lines that are different, only the lines that begin and end each set of differences are displayed. The intermediate lines are represented by ellipses (...).

The /B switch forces a binary comparison of both files. The two files are compared byte by byte, with no attempt to resynchronize after a mismatch. The mismatches are printed as follows:

```
xxxxxxxx yy zz
```

(where *xxxxxxxx* is the relative address of the pair of bytes from the beginning of the file). Addresses start at 00000000; *yy* and *zz* are the mismatched bytes from *file1* and *file2*, respectively. If one of the files contains less data than the other, a message is displayed. For example, if FILE1.TXT ends before FILE2.TXT, FC displays:

```
FC: FILE2.TXT longer than FILE1.TXT
```

This switch is the default when you compare .EXE, .COM, .SYS, .OBJ, .LIB, or .BIN files.

The /C switch causes the matching process to ignore the case of letters. All letters in the files are considered uppercase letters. For example, the following two lines would be considered matching:

```
Much MORE data IS NOT FOUND

much more data is not found
```

If both the /W and /C options are specified, then FC compresses white space and ignores case. For example, if an underscore represents a white space, then the following two lines match:

```
_____DATA_was_found_____

data_was_found
```

The /L switch compares the files in ASCII mode. This switch is the default when you compare files that do not have extensions of .EXE, .COM, .SYS, .OBJ, .LIB, or .BIN.

The /LB switch sets the internal line buffer to *n* lines. The default length of the internal buffer is 100 lines. Files that have more than this number of consecutive, differing lines will abort the comparison.

The /N switch displays the line numbers on an ASCII comparison.

The /T switch does not expand tabs to spaces. The default is to treat tabs as spaces to 8-column positions.

The /W switch causes FC to compress white space (tabs and spaces) during the comparison. If a line contains several spaces or tabs in a row, these characters are considered a single white space.

Note that although FC *compresses* white space, it does not ignore it. The exception is beginning or ending white space in a line, which is ignored. For example, of the next four examples, the first three match, but the fourth does not (note that an underscore represents a white space):

```
_ _ _More_ _data_to_be_found_ _ _ _
More_data_to_be_found
_ _ _ _ _ _More_ _ _ _ _ _data_to_be_ _ _ _found_ _ _ _ _ _
_ _ _ _Moredata_to_be_found
```

The */nnnn* switch specifies the number of lines that must match after FC finds a difference between files. If the number of matching lines in the files is less than this number, FC displays these matching lines as differences.

HOW FC REPORTS DIFFERENCES

FC reports the differences between the two files you specify by displaying the first filename, followed by the lines that differ between the files, followed by the first line to match in both files. FC then displays the name of the second file, followed by the lines that are different, followed by the first line that matches.

The default for the number of lines to match between the files is 2. (If you want to change this default, specify the number of lines with the */nnnn* switch.)

Example

```
...
...
***** file1
difference
1st matching line in file1 and file2
next matching line in file1 and file2
***** file2
difference
1st matching line in file1 and file2
next matching line in file1 and file2
...
...
```

If the number of lines in the internal buffer is less than the number of consecutive, differing lines, the program will stop. FC then displays the following message:

```
resynch failed. Files are too different.
```

FC then displays the contents of the internal buffer and returns to the MS–DOS default drive prompt (for example, A>).

REDIRECTING FC OUTPUT TO A FILE

Unless you redirect the FC output to a file, the differences and matches between the two files you specify will be displayed on your screen. You can easily redirect the FC output using the MS–DOS redirection symbols (see "REDIRECTING COMMAND INPUT AND OUTPUT" in Section 2).

To compare FILE1.TXT and FILE2.TXT and then send the FC output to DIFFER.TXT, type:

```
FC FILE1.TXT FILE2.TXT > DIFFER.TXT
```

The differences and matches between FILE1.TXT and FILE2.TXT will be put into DIFFER.TXT on the default drive.

Assume that the text files PENCIL.AD and PEN.AD exist on your disk and contain the following:

PENCIL.AD

```
    Introducing ...
    The X-1000 Automatic Pencil Sharpener
    From Sharpe Office Supplies
    The World Leader in Office Sharpeware

    This model is compact, and sharpens
    pencils more efficiently than ever before.

    Our Motto:
    "You oughta be Sharpe too!"

    I.R. Sharpe, President
```

PEN.AD

```
    Introducing ...
    The X-2000 Automatic Pen and Pencil Sharpener
    From Sharpe Office Supplies
    The World Leader in Office Sharpeware

    This model is compact, and sharpens pens and
    pencils more efficiently than ever before.

    Our Motto:
    "You oughta be Sharpe too!"

    I.R. Sharpe, President
```

Example 1

To compare the two files and display the differences on the screen, type:

```
FC PENCIL.AD PEN.AD
```

FC compares PENCIL.AD with PEN.AD and displays the differences on the screen. All other defaults remain intact. (The defaults are: perform an ASCII comparison with an internal buffer length of 100 lines, treat tabs as spaces to 8-column positions, do not ignore case, do not display line numbers, and do not compress white space.)

The following output would appear on the screen:

```
***** pencil.ad
Introducing ...
The X-1000 Automatic Pencil Sharpener
From Sharpe Office Supplies
***** pen.ad
Introducing ...
The X-2000 Automatic Pen and Pencil Sharpener
From Sharpe Office Supplies
*****

***** pencil.ad

This model is compact, and sharpens
pencils more efficiently than ever before.
***** pen.ad

This model is compact, and sharpens pens and
pencils more efficiently than ever before.
*****
```

Example 2

You can print the differences on the lineprinter using the same two files. In this example, four successive lines must be the same to constitute a match. If four such lines do not match, they are considered different and are displayed.

First, type the following command:

```
FC /4 PENCIL.AD PEN.AD > PRN
```

The following output would appear on the lineprinter:

```
***** pencil.ad
Introducing ...
The X-1000 Automatic Pencil Sharpener
From Sharpe Office Supplies
The World Leader in Office Sharpeware

This model is compact, and sharpens
pencils more efficiently than ever before.
***** pen.ad
Introducing ...
The X-2000 Automatic Pen and Pencil Sharpener
From Sharpe Office Supplies
The World Leader in Office Sharpeware

This model is compact, and sharpens pens and
pencils more efficiently than ever before.
*****
```

Example 3

This example forces a binary comparison and then displays the differences on the screen using the same two files from the previous two examples. First, enter the following:

```
FC /B PENCIL.AD PEN.AD
```

The /B switch in this example forces the binary comparison. You must type this switch and any others before the filenames in the FC command line. The following display would appear:

```
00000017 31 32
00000029 63 20
0000002A 69 61
0000002B 6C 6E
0000002C 20 64
0000002D 53 20
0000002E 68 50
0000002F 61 65
00000030 72 6E
00000031 70 63

...
...
...

0000010D 73 49
0000010E 69 2E
0000010F 64 52
00000110 65 2E
00000111 6E 20
00000112 74 53
00000113 20 68
00000114 0D 61
00000115 0A 72
00000116 1A 70
FC: PEN.AD longer than PENCIL.AD
```

Section 10

LINK: A Linker

In this section you'll learn:

- how to create executable files with LINK
- how to use LINK command options
- how LINK creates programs

INTRODUCTION

The Microsoft 8086 Object Linker (LINK) creates executable programs from object files generated by the Microsoft Macro Assembler (MASM) or by compilers for high-level languages, such as C or Pascal. The linker copies the resulting program to an executable (.EXE) output file. You can then run the program by typing the file's name on the MS-DOS command line.

To use LINK, you must create one or more object files, then submit these files, along with any required library files, to the linker for processing. LINK combines code and data in the object files and searches the named libraries to resolve external references to routines and variables. It then copies a relocatable execution image and the relocation information to the executable file. Using the relocation information, MS-DOS can load the executable image at any convenient memory location and then run it. LINK can process programs that contain up to one megabyte of code and data.

This section explains how to use the linker to create executable programs. It also defines each of the options you can use in a link command line to control the linking process. Finally, it explains how LINK creates programs.

STARTING AND USING LINK

This section explains three methods for starting and using the linker to create executable programs. These methods let you use LINK by answering a series of prompts, by typing an MS-DOS command line, or by using a response file. The three methods can also be mixed.

Once you start LINK, it will either process the files you supply or prompt you for additional files. Also, note that you can stop the linker at any time by pressing **Ctrl-C**.

Using Prompts to Specify LINK Files

When you type the command, LINK, at the MS-DOS prompt, the linker prompts you for the information it needs. Follow these steps.

1. First, type the following command and press the Enter key:

   LINK

   LINK prompts you for the object files you wish to link by displaying the following message:

   Object Modules [.OBJ]:

2. Type the name or names of the object files you wish to link. If you do not supply extensions for these files, LINK supplies .OBJ by default. If you have more than one name, make sure you separate each with spaces or plus signs (+). If you have more names than can fit on one line, type a plus sign (+) as the last character on the line and press **Enter**. LINK then prompts you for additional object files.

   Once you have given all your object filenames, press **Enter**. The linker displays the following prompt:

   Run File [*filename*.EXE]:

3. Note that in step 2, *filename* is the same as the first *filename* you typed at the Object Modules prompt. Type the name of the executable file you wish to create, and press **Enter**. If you do not give an extension, LINK supplies .EXE by default. If you want LINK to supply a default executable filename, just press **Enter**. The *filename* will then be the same as the first object file, but with the extension .EXE.

   Once you have pressed **Enter**, LINK displays the following prompt:

   List File [NUL.MAP]:

4. Type the name of the map file you wish to create, then press the **Enter** key. If you do not supply a filename extension, the linker uses .MAP by default. If you don't want a map file, don't type a filename. Just press **Enter**.

   Once you have pressed the **Enter** key, LINK displays the following prompt:

   Libraries [.LIB]:

5. Type the names of any library files containing routines or variables referenced but not defined in your program. If you give more than one name, make sure the names are separated by spaces or plus signs (+). If you don't supply filename extensions, the linker uses .LIB by default. If you have more names than can fit on one line, type a plus sign (+) as the last character on the line and press **Enter**. LINK then prompts you for additional filenames.

   After entering all names, press the **Enter** key. If you don't want to search any libraries, don't type any names; just press **Enter**.

LINK now creates the executable file.

When entering filenames, you must supply a pathname for any file that is not in the current drive and directory. You can use LINK options by typing them after the filename at any prompt. If the linker cannot find an object file, it displays a message and waits for you to change disks, if necessary.

At any prompt, you can type the rest of the filenames by using the command line format described in the next section, "Using a Command Line to Specify LINK Files." For example, you can choose the default responses for all remaining prompts by typing a semicolon (;) after any prompt, or you can type commas (,) to indicate several files. (If you type a semicolon at the Object Modules prompt, be sure to supply at least one object filename.) When the linker encounters a semicolon, it immediately chooses the default responses and processes the remaining files without displaying any more prompts.

Example

The following example links the object modules MODA.OBJ, MODB.OBJ, MODC.OBJ, and STARTUP.OBJ; searches the library file MATH.LIB on drive B of the \LIB directory for routines and data used in the program; and creates an executable file named MODA.EXE, and a map file named ABC.MAP. The /pause option in the Object Modules prompt line then causes LINK to pause while you change disks, after which the linker creates the executable file (see the section entitled "Pausing to Change Disks," later in this section):

```
LINK

Object Modules [.OBJ]: MODA+MODB+
Object Modules [.OBJ]: MODC+STARTUP/PAUSE
Run File [moda.EXE]:
List File [NUL.MAP]: ABC
Libraries [.LIB]: B:\LIB\MATH
```

Using a Command Line to Specify LINK Files

You can create an executable program by typing LINK, followed by the names of the files you wish to process. The command line has the following general form:

LINK *objectfiles* [,[*executablefile*][,[*mapfile*] [,[*libraryfile*]]]]
[*options*][;]

The variables in this command line are described as follows:

objectfiles

Includes the name or names of object files that you want to link together. The files must have been created using MASM or a high-level-language compiler. The linker requires at least one object file. If you do not supply an extension, LINK provides the extension .OBJ.

executablefile

Is an optional placeholder for the name you wish to give the executable file that LINK will create. If you do not supply an executable file, LINK creates a filename by using the name of the first object file in the command line and appending it with an .EXE extension.

mapfile

Is the name of the file that receives the map listing. If you do not supply an extension, the linker provides the extension .MAP. If you specify the /MAP or /LINENUMBERS option, the linker creates a map file even if you do not specify one in your command line.

libraryfiles

Includes the name or names of the libraries containing routines that you wish to link to create a program. If you do not supply an extension, LINK supplies the extension .LIB.

options

Controls the operation of LINK. You can use any of the options listed in the section entitled “The LINK Options.” You can specify options anywhere on the command line.

If you do not specify a drive or a directory for a file, LINK assumes the file is on the current drive and directory. You cannot specify the drive and the directory for the *objectfile* and expect LINK to supply the same drive and directory for the other files. Instead, you must give the location of each file specifically.

If you type the comma after the object file, LINK supplies the default name for the *executablefile* and suppresses the *mapfile* and *libraryfiles*. You can also use a semicolon (;) anywhere after the object file to terminate the command line.

If you do not supply all filenames in the command line and do not end the command line with a semicolon, the linker prompts you for additional files, using the prompts described previously in the section "Using Prompts to Specify Link Files." If you give more than one object file or library file, you must separate the names by spaces or plus signs (+).

If you do not specify a drive or directory for a file, LINK assumes the file is on the current drive and directory. You cannot specify the drive or directory for the *objectfile* and expect LINK to supply the same drive and directory for other files. Instead, you must give the location of each file specifically.

When you are linking modules produced by a high-level language compiler that supports overlays, you must specify overlay modules by putting them in parentheses. Since MASM has no overlay manager, you can specify overlays only for object files linked with the run-time library of a language compiler that supports overlays.

For example, you can use overlays with modules compiled with Microsoft FORTRAN, versions 3.2 and later, Microsoft Pascal versions 3.2 and later, and Microsoft C versions 3.0 and later. See your language compiler's manual for details on specifying overlays.

Examples

The first example below uses the object file file.OBJ to create the executable file FILE.EXE. LINK searches the FILE.LIB library for routines and variables used within the program. It also creates a file called FILE.MAP, which contains a list of the program's segments and groups:

```
LINK FILE.OBJ,FILE.EXE,FILE.MAP,ROUTINE.LIB
```

The first example is equivalent to the following line:

```
LINK FILE,,ROUT!NE
```

The second example uses the two object files, STARTUP.OBJ and FILE.OBJ, on the current drive to create an executable file named

FILE.EXE on drive B. LINK creates a map file on the \map directory of the current drive, but does not search any libraries:

```
LINK STARTUP+FILE,B:FILE,\MAP\FILE;
```

The final example links the object modules MODA.OBJ, MODB.OBJ, MODC.OBJ, and STARTUP.OBJ:

```
LINK MODA MODB MODC STARTUP/PAUSE,,ABC,B:\LIB\MATH
```

The linker searches through the library file MATH.LIB in the \LIB directory on drive B for routines and data used in the program. It then creates an executable file named MODA.EXE, and a map file named ABC.MAP.

The pause option in the command line causes the linker to pause and ask you to change disks before it creates the executable file. (This option is described in more detail in "PAUSING TO CHANGE DISKS" later in this section.)

Using A Response File to Specify Link Files

You can create a program by listing, in a response file, the names of all the files to be processed, and by giving the name of the response file on the LINK command line. The simplest way to use a response file is with a command line of the following form:

LINK *@filename*

You can also specify a response file at any prompt, or at any position in a command line. The input from the response file is treated exactly as though you had typed it at the LINK prompts or in a command line. However, any Return/linefeed combinations in the file are treated the same as if you had pressed the Enter key in response to a prompt, or typed a comma in a command line.

When you specify a response file, remember that the filename must be the name of the response file, and that you must precede it by an "at" sign (@). If the file is in another directory or on another disk drive, you must provide a pathname.

You can name the response file anything you like. The file content has the following general form:

objectfiles
[*executablefile*]
[*mapfile*]
[*libraryfiles*]

You can omit any elements that have already been provided at prompts or with a partial command line.

You must place each group of filenames on a separate line. If you have more names than can fit on one line, you can simply continue the names on the next line by typing a plus sign (+) as the last character in the current line and pressing the Enter key. If you do not supply a filename for a group, you must leave an empty line. You can give options on any line.

You can place a semicolon (;) on any line in the response file. When LINK encounters the semicolon, it automatically supplies default filenames for all files you have not yet named in the response file. The remainder of the response file is ignored.

When you create a program with a response file, the linker displays each response from your response file on the screen in the form of prompts. If the response file does not contain names for required files, LINK prompts you for the missing names and waits for you to enter responses. A response file should end with either a semicolon (;) or a Return/linefeed combination. If you fail to provide a final Return/linefeed in the file, the linker will display the last line of the response file and wait for you to press the Enter key.

Example

The following response file tells the linker to link the four object modules, MODA, MODB, MODC, and STARTUP. Then, before producing the executable file MODA.EXE, it tells LINK to pause to let you swap disks. Finally, the linker creates a map file ABC.MAP and searches the MATH.LIB library in the \LIB directory of drive B:

```
MODA MODB MODC STARTUP /PAUSE
ABC
B:\LIB\MATH
```

The following procedure combines all three methods of supplying filenames. Assume you have a response file called library that contains one line:

```
LIB1+LIB2+LIB3+LIB4
```

Now start LINK with a partial command line:

```
LINK OBJECT1 OBJECT2
```

LINK takes OBJECT1.OBJ and OBJECT2.OBJ as its object files, and prompts for the next file:

```
Run File [OBJECT1.EXE]: EXEC
List File [NUL.MAP]:
Libraries [.LIB]: @LIBRARY
```

You include the name EXEC so that the linker will name the executable file EXEC.EXE. You then press **Enter** to indicate that no map file is desired, and you enter *@LIBRARY* so that the linker will read in the response file containing the four library filenames.

Using Search Paths with Libraries

You can direct LINK to search directories and disk drives for the libraries you have named in a command by either specifying one or more search paths with the library names, or by assigning the search paths to the environment variable LIB before you invoke LINK. Environment variables are explained under the SET command in Section 4, "MS–DOS Commands."

A search path is the path of a directory or drive name. You type search paths along with library names on the LINK command line or in response to the Libraries prompt. You can also specify up to 16 search paths and assign them to the LIB environment variable by using the MS–DOS set command. In the latter case, you must separate the search paths by semicolons (;).

If you include a drive or directory name in the filename for a library in the LINK command line, the linker searches there only. If you don't give a drive or directory name, LINK searches for library files in the following order:

1. First, the linker searches the current drive and directory.
2. If the library is not found and one or more search paths have been given in the command line, the linker searches the specified search paths in the order in which you gave them.
3. If the library is still not found and you have set a search path by using the LIB environment variable, the linker searches there.
4. If the library is still not found, LINK prints an error message.

Examples

In the first example, the linker searches only the \ALTLIB directory on drive A to find the MATH.LIB library. To find COMMON.LIB, it will search the current directory on the current drive, the current directory on drive B, and finally the \LIB directory on drive D:

```
LINK FILE,,FILE,A:\ALTLIB\MATH.LIB+COMMON+B:+D:\LIB\
```

In the second example, LINK searches the current directory, the \LIB directory on drive C, and the \SYSTEM\LIB directory on drive U to find the libraries, MATH.LIB and COMMON.LIB:

```
SET LIB=C:\LIB;U:\SYSTEM\LIB
LINK FILE,,FILE.MAP,MATH+COMMON
```

The Map File

The map file lists the names, load addresses, and lengths of all segments in a program. It also lists the names and load addresses of any groups in the program, the program start address, and messages about any errors it may have encountered. If the /map option is used in the LINK command line, the map file lists the names and load addresses of all public symbols.

Segment information has the general form shown in this example:

```
Start       Stop        Length      Name       Class

00000H      0172CH      0172DH      TEXT       CODE
01730H      01E19H      006EAH      DATA       DATA
```

The Start and Stop columns show the 20-bit addresses (in hexadecimal) of the first and last byte in each segment. These addresses are relative to the beginning of the load module, which is assumed to be address 0000H. The operating system chooses its own starting address once the program is actually loaded. The Length column gives the length of the segment in bytes; the Name column gives the name of segment: and the Class column gives the segment's class name.

Group information has the following general form:

```
Origin      Group

0000:0      IGROUP
0173:0      DGROUP
```

In this example, IGROUP is the name of the code (instruction) group and DGROUP is the name of the data group.

At the end of the listing file, the linker gives you the address of the program entry point.

If you specify the /MAP option in the LINK command line, the linker adds a public-symbol list to the map file. The symbols are presented twice: once in alphabetical order, then in the order of their load addresses. The list has the general form shown in the following example:

```
Address        Publics by Name

0000:1567      BRK
0000:1696      CHMOD
0000:01DB      CHKSTK
0000:131C      CLEARERR
0173:0035      FAC

Address        Publics by Value

0000:01DB      CHKSTK
0000:131C      CLEARERR
0000:1567      BRK
0000:1696      CHMOD
0000:0035      FAC
```

The addresses of the public symbols are in segment:offset format. They show the location of the symbol relative to the beginning of the load module, which is assumed to be at address 0000:0000.

When the /HIGH AND /DSALLOCATE options are used and the program's code and data combined do not exceed 64K bytes, the map file may show symbols that have unusually large segment addresses. These addresses indicate a symbol whose location is below the actual start of the program code and data.

For example, the following symbol entry shows that TEMPLATE is located below the start of the program:

```
FFF0:0A20     TEMPLATE
```

Note that the 20-bit address of TEMPLATE is 00920H.

The Temporary Disk File--VM.TMP

LINK normally uses available memory for the LINK session. If it runs out of available memory, it creates a temporary disk file named VM.TMP in the current working directory. When the linker creates this file, it displays the following message:

```
VM.TMP has been created.
Do not change diskette in drive x:
```

After this message appears, you must not remove the disk from the drive specified by *x* until the link session ends. The /PAUSE option cannot be used if a temporary file is created. After LINK has created the executable file, it deletes the temporary file automatically.

Do not use the VM.TMP filename for your own files, since when the linker creates the temporary file, it destroys any previous file that has the same name.

THE LINK OPTIONS

The linker options specify and control the tasks that LINK performs. All options begin with the linker-option character, which is a slash (/). You can use the following options anywhere on a LINK command line:

OPTION NAME	ACTION
/HELP	Shows the list of options
/PAUSE	Pauses during linking
/EXEPACK	Packs executable files
/MAP	Creates a public-symbol map
/LINENUMBERS	Copies line numbers to a map file
/NOIGNORECASE	Preserves case sensitivity in names
/NODEFAULTLIBRARYSEARCH	Overrides default libraries
/STACK	Sets stack size
/CPARMAXALLOC	Sets maximum allocation space
/HIGH	Sets a high load address for a program
/DSALLOCATE	Allocates a data group
/NOGROUPASSOCIATION	Sets a group association override
/OVERLAYINTERRUPT	Sets an overlay interrupt
/SEGMENTS	Sets a maximum number of segments
/DOSSEG	Specifies MS-DOS segment ordering

You can abbreviate an option name as long as your abbreviation contains enough letters to distinguish the specified option from other options. Minimum abbreviations are listed for each option.

Many of the LINK options set values in the MS-DOS program header. You will understand these options better if you understand how the header is organized. The program header is described in the *MS-DOS Programmer's Reference* and in some reference books on MS-DOS.

VIEWING THE OPTIONS LIST

Syntax

/HELP

The /HELP option causes LINK to write a list of the available options to the screen. If you ever need a reminder of the available options, you may find this list convenient. You should not give a filename when using the /HELP option.

Minimum abbreviation: /HE

Example

```
LINK /HELP
```

PAUSING TO CHANGE DISKS

Syntax

/PAUSE

The /PAUSE option causes LINK to pause before writing the executable file to disk so that you can swap disks before the linker writes the executable (.EXE) file to disk.

If you specify the /PAUSE switch, the linker displays the following message before creating the run file:

```
About to generate .EXE file
Change diskette in drive x: and press <ENTER>
```

Note that *x*: is the proper drive name. This message appears after the linker has read data from the object files and library files, and after it has written data to the map file, if you specified one. LINK resumes processing when you press the **Enter** key, and after it writes the executable file to disk, it displays the following message:

```
Please replace original diskette
in drive letter and press <ENTER>
```

Minimum abbreviation: /P

Do not remove the disk used for the VM.TMP file, if such a file has been created. If the temporary disk message appears when you have specified the

/PAUSE option, you should press **Ctrl–C** to terminate the LINK session. Rearrange your files so that the temporary file and the executable file can be written to the same disk, then try again.

Example

The following command causes the linker to pause just before creating the executable file FILE.EXE. After creating this file, LINK pauses again to let you replace the original disk:

```
LINK FILE/PAUSE,FILE,,\LIB\MATH
```

PACKING EXECUTABLE FILES

Syntax

/EXEPACK

The /EXEPACK option directs LINK to remove sequences of repeated bytes (typically nulls) and optimize the loadtime relocation table before creating the executable file. Executable files linked with the /EXEPACK option may be smaller, and, thus, load faster than files linked without the option. However, the Microsoft Symbolic Debug Utility (SYMDEB) cannot be used with packed files.

The /EXEPACK option does not always save a significant amount of disk space (in some cases it may even increase file size). Programs that have a large number of loadtime relocations (about 500 or more) and long streams of repeated characters will usually be shorter if packed. If you are not sure if your program meets these conditions, try linking it both ways and compare the results.

Minimum abbreviation: /E

Example

This example creates a packed version of the file program.EXE:

```
LINK PROGRAM /E;
```

PRODUCING A PUBLIC-SYMBOL MAP

Syntax

/MAP

The /MAP option causes LINK to produce a listing of all public symbols declared in your program. This list is copied to the map file that LINK creates. For a complete description of the listing-file format, see the section, "The Map File," earlier in this section. The /MAP option is required if you want to use symdeb for symbolic debugging.

Minimum abbreviation: /M If you do not specify a map file in a LINK command, you can use the /MAP option to force the linker to create one. LINK gives the forced map file the same filename as the first object file specified in the command. It also adds the default extension .MAP.

Example

The following command creates a map of all public symbols in the file FILE.OBJ:

```
    LINK FILE,,/MAP;
```

COPYING LINE NUMBERS TO THE MAP FILE

Syntax

/LINENUMBERS

The /LINENUMBERS option directs the linker to copy the starting address of each program source line to a map file. The starting address is actually the address of the first instruction that corresponds to the source line. You can use the MAPSYM program to copy line-number data to a symbol file, which can then by used by SYMDEB.

The linker copies the line-number data only if you give a map-file name in the LINK command line, and only if the given object file has line-number information. Line numbering is available in some high-level-language compilers, including Microsoft FORTRAN and Pascal, versions 3.0 and later, and Microsoft C, versions 2.0 and later.

MASM does not copy line-number information to the object file. If an object file has no line-number information, the linker ignores the /LINENUMBERS option.

Minimum abbreviation: /LI

If you do not specify a map file in a LINK command, you can still use the /LINENUMBERS option to force the linker to create one. Just place the option at or before the List File prompt. LINK gives the forced map file the same filename as the first object file that you specified in the command, and gives it the default extension .MAP.

Example

This example causes the line–number information in the object file FILE.OBJ to be copied to the map file FILE.MAP:

```
LINK FILE/LINENUMBERS,,EM+SLIBFP
```

PRESERVING LOWERCASE

Syntax

/NOIGNORECASE

The /NOIGNORECASE option directs LINK to treat uppercase and lowercase letters in symbol names as distinct letters. Normally, LINK considers uppercase and lowercase letters to be identical, treating the words "TWO", "two", and "Two" as the same symbol. When you use the /NOIGNORECASE option, however, the linker treats "TWO", "two", and "Two" as different symbols.

Typically, you use the /NOIGNORECASE option with object files created by high–level–language compilers. Some compilers treat uppercase and lowercase letters as distinct letters and assume the linker does the same.

If you are linking modules created with MASM to modules created with a case–sensitive language such as C, make sure public symbols have the same sensitivity in both modules. For example, you could make all variables in C distinctive by spelling, regardless of case, and then LINK without the /NOIGNORECASE option. Another alternative would be to use the /ML or MX option to make public variables in MASM case–sensitive. Then LINK with the /NOIGNORECASE option.

Minimum abbreviation: /NOI

Example

The following command causes the linker to treat uppercase and lowercase letters in symbol names as distinct letters. The object file FILE.OBJ is linked with routines from the standard C language library \SLIBC.LIB located in the \LIB directory. The C language expects uppercase and lowercase letters to be treated distinctly:

```
LINK FILE1+FILE2/NOI,,,EM+MLIBFP
```

IGNORING DEFAULT LIBRARIES

Syntax

/NODEFAULTLIBRARYSEARCH

The /NODEFAULTLIBRARYSEARCH option directs the linker to ignore any library names it may find in an object file. A high–level–language compiler may add a library name to an object file to ensure that a default set of libraries is linked with the program. Using this option overrides these default libraries and lets you explicitly name the libraries you want by including them on the LINK command line.

Minimum abbreviation: /NOD

Example

The following example links the object files, STARTUP.OBJ and FILE.OBJ, with routines from the libraries, EM, SLIBFP, and SLIBC. Any default libraries that may have been named in STARTUP.OBJ or FILE.OBJ are ignored:

```
LINK STARTUP+FILE/NOD,,,EM+SLIBFP+SLIBC
```

SETTING THE STACK SIZE

Syntax

/STACK:*size*

The /STACK option sets the program stack to the number of bytes given by size. The linker usually calculates a program's stack size automatically, basing it on the size of any stack segments given in the object files. If you do use the /STACK option, the linker uses the value you type in place of any value it may have calculated.

The size can be any positive integer in the range from 1 to 65535. This value can be a decimal, octal, or hexadecimal number. Octal numbers must begin with a zero, and hexadecimal numbers must begin with a leading zero followed by a lowercase x. For example, 0x1B.

By using the EXEMOD utility, you can also change the stack size after linking.

Minimum abbreviation: /ST

Examples

The first example sets the stack size to 512 bytes:

```
LINK FILE/STACK:512,,;
```

The second example sets the stack size to 255 (FFH) bytes:

```
LINK MODA+MODB,RUN/ST:0xFF,AB,\LIB\START;
```

The final example sets the stack size to 24 (30 octal) bytes:

```
LINK STARTUP+FILE/ST:030,,;
```

SETTING THE MAXIMUM ALLOCATION SPACE

Syntax

/CPARMAXALLOC:*number*

The /CPARMAXALLOC option sets the maximum number of 16-byte paragraphs needed by a program when it is loaded into memory. The operating system uses this number when allocating space for a program prior to loading it.

LINK normally sets the maximum number of paragraphs to 65535. Since this represents all addressable memory, the operating system always denies the default setting and allocates the largest contiguous block of memory it can find. If you use the /CPARMAXALLOC option, the operating system allocates no more space than is given by this option. This means any additional space in memory is free for other programs.

The *number* can be any integer in the range from 1 to 65535. It must be a decimal, octal, or hexadecimal number. Octal numbers must begin with a zero, and hexadecimal values must begin with a leading zero followed by a lowercase x. For example, 0x2B.

If *number* is less than the minimum number of paragraphs needed by the program, LINK ignores your request and sets the maximum value equal to the minimum needed. The minimum number of paragraphs needed by a program is never less than the number of paragraphs of code and data in the program.

Minimum abbreviation: /C

Examples

The first example sets the maximum allocation to 15 paragraphs:

```
LINK FILE/C:15,,;
```

The second example sets the maximum allocation to 255 (FFH) paragraphs:

```
LINK MODA+MODB,RUN/CPARMAXALLOC:0xFF,AB;
```

The final example sets the maximum allocation to 24 (30 octal) paragraphs:

```
LINK STARTUP+FILE,/C:030,;
```

SETTING A HIGH START ADDRESS

Syntax

/HIGH

The /HIGH option sets a program's starting address to the highest possible address in free memory. If you don't use the /HIGH option, LINK sets the program's starting address as low as possible in memory.

Minimum abbreviation: /H

Example

This example sets the starting address of the program in FILE.EXE to the highest possible address in free memory:

```
LINK STARTUP+FILE/HIGH,,;
```

ALLOCATING A DATA GROUP

Syntax

/DSALLOCATE

The /DSALLOCATE option directs the linker to reverse its normal processing when assigning addresses to items belonging to the group named DGROUP. Normally, LINK assigns the offset 0000H to the lowest byte in a group. If you use /DSALLOCATE, LINK assigns the offset FFFFH to the highest byte in the group. The result is data that appear to be loaded as high as possible in the memory segment containing DGROUP. Typically, you use the /DSALLOCATE option with the /HIGH option to take advantage of unused memory before the start of the program. The linker assumes that all free bytes in DGROUP occupy the memory preceding the program. To use the group, you must set a segment register to the start address of DGROUP.

Minimum abbreviation: /D

Example

The following example directs the linker to place the program as high in memory as possible, then adjust the offsets of all data items in DGROUP so that they are loaded as high as possible within the group:

```
LINK STARTUP+FILE/HIGH/DSALLOCATE,,,EM+MLIBFP
```

REMOVING GROUPS FROM A PROGRAM

Syntax

/NOGROUPASSOCIATION

The /NOGROUPASSOCIATION option directs LINK to ignore group associations when assigning addresses to data and code items.

Minimum abbreviation: /NOG

This option exists strictly for compatibility with older versions of FORTRAN and Pascal (Microsoft versions 3.13 or earlier, or any IBM version prior to 2.0). You should never use the /NOGROUPASSOCIATION option except to link with object files produced by those compilers, or with the run–time libraries that accompany the old compilers.

SETTING THE OVERLAY INTERRUPT

Syntax

/OVERLAYINTERRUPT:*number*

The /OVERLAYINTERRUPT option sets the interrupt number of the overlay loading routine to *number*. This option overrides the normal overlay interrupt number (03FH).

Number can be any integer value in the range from 0 to 255. It must be a decimal, octal, or hexadecimal number. Octal numbers must have a leading zero, and hexadecimal numbers must start with a leading zero followed by a lowercase x. For example, 0x3B. MASM does not have an overlay manager. Therefore, you can use this option only if you are linking with a run-time module from a language compiler that supports overlays. Check your compiler documentation, since you may not be able to use this option with some compilers.

Minimum abbreviation: /O

You should not use interrupt numbers that conflict with the standard MS–DOS interrupts.

Examples

The first example sets the overlay interrupt number to 255:

```
LINK FILE/O:255,,,87+SLIBFP
```

The second example sets the overlay interrupt number to 255 (FFH):

```
LINK MODA+MODB,
RUN/OVERLAY:0xFF,AB.MAP,EM+MLLIBFP/F>
```

The final example sets the overlay interrupt number to 255 (377 octal):

```
LINK STARTUP+FILE,/O:0377,,EM+MLIBFP
```

SETTING THE MAXIMUM NUMBER OF SEGMENTS

Syntax

/SEGMENTS:*number*

The /SEGMENTS option directs the linker to process no more than number segments per program. If it encounters more than the given limit, the linker displays an error message, and stops linking. You use this option to override the default limit of 128 segments.

If you do not use /SEGMENTS, the linker allocates enough memory space to process up to 128 segments. If your program has more than 128 segments, you will need to set the segment limit higher to increase the number of segments that LINK can process. If you get the following LINK error message, you should set the segment limit lower:

```
Segment limit set too high
```

The *number* can be any integer value in the range from 1 to 1024. It must be a decimal, octal, or hexadecimal number. Octal numbers must have a leading zero, and hexadecimal numbers must start with a leading zero followed by a lowercase x. For example, 0x4B.

Minimum abbreviation: /SE

Examples

The first example sets the segment limit to 192:

```
LINK FILE/SE:192,,;
```

The second example sets the segment limit to 255 (FFH).

```
LINK MODA+MODB,RUN/SEGMENTS:0xFF,AB,EM+MLIBFP;
```

USING DOS SEGMENT ORDER

Syntax

/DOSSEG

The /DOSSEG option causes LINK to arrange all segments in the executable file according to the MS–DOS segment–ordering convention. This convention has the following rules:

- All segments having the class name CODE are placed at the beginning of the executable file.
- Any other segments that do not belong to the group, DGROUP, are placed immediately after the CODE segments.
- All segments belonging to DGROUP are placed at the end of the file. If you do not use the /DOSSEG option, see "Order of Segments," later in this section, for an explanation of the normal segment order.

Minimum abbreviation: /DO

Example

The following command causes the linker to create an executable file named FILE.EXE whose segments are arranged according to the MS-DOS segment-ordering convention. The segments in the object files START.OBJ and TEST.OBJ and any segments copied from the libraries MATH.LIB and COMMON.LIB are arranged according to the same segment-ordering convention as above.

```
LINK START+TEST/DOSSEG,,,MATH+COMMON
```

HOW LINK WORKS

LINK creates an executable file by concatenating a program's code and data segments according to the instructions in the original source files. These concatenated segments form an *executable image* that is copied directly into memory when you run the program. The order and manner in which the linker copies segments to the executable file defines the order and manner in which it loads the segments into memory.

You can tell the linker how to link a program's segments by using a SEGMENT directive to supply segment attributes, or by using the GROUP directive to form segment groups. These directives define group associations, classes, and align and combine types that define the order and relative starting addresses of all segments in a program. This information works in addition to any information you supply through command line options.

The following sections explain the process that LINK uses to concatenate segments and resolve references to items in memory.

Alignment of Segments

The linker uses a segment's align type to set the starting address for the segment. The align types are *byte*, *word*, *para*, and *page*. These types correspond to starting addresses at byte, word, paragraph, and page boundaries, representing addresses that are multiples of 1, 2, 16, and 256, respectively. The default align type is *para*.

When the linker encounters a segment, it checks the align type before copying the segment to the executable file. If the align type is *word*, *para*, or *page*, the linker checks the executable image to see if the last byte copied ends at an appropriate boundary. If it doesn't, LINK pads the image with extra null bytes.

Frame Number

The linker computes a starting address for each segment in a program. The starting address is based on a segment's align type and on the size of the segments already copied to the executable file. The address consists of an offset and a canonical frame number, which specifies the address of the first paragraph in memory that contains one or more bytes of the segment. A frame number is always a multiple of 16 (a paragraph address), and the offset is the number of bytes from the start of the paragraph to the first byte in the segment. For *byte* and *word* align types, the offset may be nonzero, but the offset is always zero for *para* and *page* align types.

The frame number of a segment can be obtained from a LINK file. The frame number is the first five hexadecimal digits of the start address specified for the segment.

Order of Segments

LINK copies segments to the executable file in the same order that it encounters them in the object files. The linker maintains this order throughout the program unless it encounters two or more segments with the same class name. Segments with identical class names belong to the same class type, and are copied to the executable file as contiguous blocks.

The *Microsoft Macro Assembler Reference Manual* includes a more detailed discussion of segment loading order and methods of controlling loading order by assigning class types.

Combined Segments

LINK uses combine types to determine whether two or more segments sharing the same name should be combined into a single large segment. The combine types are *public*, *stack*, *common*, *memory*, *at*, and *private*. Combine types are also described in the *Microsoft Macro Assembler Reference Manual.*

If a segment has a *public* combine type, the linker automatically combines it with any other segments that have the same name and belong to the same class. When LINK combines segments, it ensures that the segments are contiguous and that all addresses in the segments can be accessed using an offset from the same frame address. The result is the same as if the segment were defined as a whole in the source file.

The linker preserves each segment's align type. This means that even though the segments belong to a single, large segment, the code and data in the segments retain their original align type. If the combined segments exceed 64K bytes, LINK displays an error message.

If a segment has a *stack* combine type, the linker carries out the same combine operation as for public segments. The only difference is that stack segments cause LINK to copy an initial stack–pointer value to the executable file. This stack–pointer value is the offset to the end of the first stack segment (or combined stack segment) that the linker encounters.

If you use the *stack* type for stack segments, you do not need to give instructions to load the segment into the SS register.

If a segment has a *common* combine type, the linker combines it automatically with any other segments that have the same name and belong to the same class. When LINK combines common segments, however, it places the start of each segment at the same address, creating a series of overlapping segments. The result is a single segment no larger than the largest of the combined segments.

The linker treats segments with *memory* combine types exactly like segments with *public* combine types. The Microsoft Macro Assembler (MASM), provides combine type memory for compatibility with linkers that support a separate combine type for memory segments.

A segment has a *private* combine type only if no explicit combine type is defined for it in the source file. LINK does not combine private segments.

Groups

Groups permit noncontiguous segments that do not belong to the same class to be addressable relative to the same frame address. When LINK encounters a group, it adjusts all memory references to items in the group so that they are relative to the same frame address.

Segments in a group do not have to be contiguous and do not have to belong to the same class. Nor do they have to have the same combine type. The only requirement is that all segments in the group fit within 64K bytes.

Groups do not affect the order in which the segments are loaded. Unless you use class names and enter object files in the right order, there is no guarantee that the segments will be contiguous. In fact, the linker may place segments that do not belong to the group in the same 64K bytes of memory. Although LINK does not explicitly check whether all segments in a group fit within this 64K of memory, the linker is likely to encounter a *fixup–overflow* error if this requirement is not met.

Groups, and how to define them, are discussed further in the *Microsoft Macro Assembler Reference Manual.*

Fixups

Once the starting address of each segment in a program is known, and all segment combinations and groups have been established, the linker can fix up any unresolved references to labels and variables. To fix up unresolved references, the linker computes an appropriate offset and segment address and replaces the temporary values, generated by the assembler, with the new values.

LINK carries out fixups for four different references:

- Short
- Near self-relative
- Near segment-relative
- Long

The size of the value to be computed depends on the type of reference. If LINK discovers an error in the anticipated size of a reference, it displays a *fixup-overflow* message. This error can happen, for example, if a program attempts, by using a 16-bit offset, to reach an instruction in a segment that has a different frame address. The error can also occur if the segments in a group do not fit within a single 64K-byte block of memory.

A short reference occurs in JMP instructions that attempt to pass control to labeled instructions in the same segment or group. The target instruction must be no more than 128 bytes from the point of reference. The linker computes a signed, 8-bit number for this reference and displays an error message if the target instruction belongs to a different segment or group (that is, if it has a different frame address), or if the target is more than 128 bytes distant (in either direction).

A near self-relative reference occurs in instructions which access data relative to the same segment or group. The linker computes a 16-bit offset for this reference and displays an error message if the data are not in the same segment or group.

A near segment-relative reference occurs in instructions that attempt to access data in a specified segment or group, or that is relative to a specified segment register. LINK computes a 16-bit offset for this reference and displays an error message if either of the following conditions exists: the offset of the target within the specified frame is greater than 64K bytes or less than 0, or the beginning of the canonical frame of the target is not addressable.

A long reference occurs in CALL instructions that attempt to access an instruction in another segment or group. LINK computes a 16-bit frame address and 16-bit offset for this reference and displays an error message if either of the following conditions exists: the computed offset is greater than 64K bytes or less than 0, or the beginning of the canonical frame of the target is not addressable. canonical frame of the target is not addressable.

Section 11

DEBUG

In this section you'll learn:

- how to start the DEBUG utility
- how to use the DEBUG commands and parameters.

INTRODUCTION

The DEBUG utility is a debugging program that provides a controlled testing environment for binary and executable object files. Note that EDLIN, the MS-DOS line editor, is used to alter source files; DEBUG is EDLIN's counterpart for binary files.

DEBUG eliminates the need to reassemble a program to see if a problem has been fixed by a minor change. It allows you to alter the contents of a file or the contents of a CPU register, and then immediately reexecute a program to check the validity of the changes made.

All DEBUG commands may be aborted at any time by pressing Ctrl-C. The Ctrl-S key sequence suspends the display, so that you can read it before the output scrolls away. Pressing any key other than Ctrl-C or Ctrl-S restarts the display. All these commands are consistent with the control character functions available at the MS-DOS command level.

HOW TO START DEBUG

DEBUG may be started two ways. By the first method, you type all commands in response to the DEBUG prompt (a hyphen). By the second method, you type all commands on the line used to start DEBUG.

Method 1: **DEBUG**
Method 2: **DEBUG** [*filename* [*arglist*]]

Method 1: DEBUG

To start DEBUG using method 1, simply type the following:

```
DEBUG
```

DEBUG responds with the hyphen (-) prompt, signaling that it is ready to accept your commands. Since you didn't specify a filename, you can use other commands to work on current memory, disk sectors, or disk files. CAUTION

When DEBUG (version 2.0) is started, it sets up a program header at offset 0 in the program work area. In previous versions of DEBUG, you could overwrite this header. You can still overwrite the default header if you don't give a filename to DEBUG. If you are debugging a .COM or .EXE file, however, do not tamper with the program header below address 5CH, or DEBUG will terminate.

Do not restart a program after the following message is displayed:

```
Program terminated normally
```

You must reload the program with the N (NAME) and L (LOAD) commands for it to run properly.

Method 2: Command Line

To start DEBUG using a command line, you must use the following syntax:

DEBUG [*filename* [*arglist*]]

For example, if you specify a filename, the following would be a typical command to start DEBUG:

DEBUG *file*.**EXE**

DEBUG would then load file.exe into memory starting at 100 (hexadecimal) in the lowest available segment. The BX:CX registers are loaded with the number of bytes placed into memory.

If you do include a filename, you might also specify an arglist. An arglist is a list of filename parameters and switches that are to be passed to the program filename. So when filename is loaded into memory, it is loaded as if it had been started with a command of the form, DEBUG filename arglist.

Here, filename is the file to be debugged, and arglist is the rest of the command line used when DEBUG calls and loads filename into memory.

DEBUG COMMAND INFORMATION

Each DEBUG command consists of a single letter followed by one or more parameters. Additionally, the control characters and special editing functions described in Section 6, "MS-DOS Editing and Function Keys," apply to DEBUG as well.

If a syntax error occurs in a DEBUG command, DEBUG reprints the command line and indicates the error with a caret (^) and the word "Error" as in the following example:

```
    dcs: 100 cs:110
          ^ Error
```

Note that when typing commands and parameters you may use any combination of uppercase and lowercase letters.

The DEBUG commands are listed in Table 11-1.

Table 11-1 DEBUG Commands

DEBUG 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

DEBUG COMMAND PARAMETERS

All DEBUG commands accept parameters, except the Q (QUIT) command. Parameters may be separated by delimiters (spaces or commas), but a delimiter is required only between two consecutive hexadecimal values. Thus, the following commands are equivalent:

```
dcs:100 110
d cs:100 110
d,cs:100,110
```

Command parameters are listed below.

PARAMETER — DEFINITION

drive: — A one-digit hexadecimal value that indicates which drive a file will be loaded from or written to. The valid values are 0–3, where 0=A:, 1=B:, 2=C:, 3=D:.

byte — A two-digit hexadecimal value placed in or read from an address or register.

record — 1-digit to 3-digit hexadecimal value that indicates the logical record number on the disk and the number of disk sectors to be written or loaded. Logical records correspond to sectors; however, since they represent the entire disk space, their numbering differs.

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 containing either an alphabetic segment register or a four-digit segment address plus an offset value. You may omit the segment name or segment address, in which case the default segment DS is used for all commands except G, L, T, U, and W, for which the default segment is CS. All numeric values are hexadecimal.

Following is an example *address*:

```
CS:0100
04BA:0100
```

Note that the colon is required between the segment name (whether numeric or alphabetic) and the offset value.

range

Contains two addresses: for example, *address address*; or one address, an L, and a value: for example, *address* L *value* where *value* is the number of lines on which the command should operate (L80 is assumed). The second type of *range* cannot be used if another hexadecimal value follows, since the hexadecimal value would be interpreted as the second *address* of the *range*.

Here are some example ranges:

```
CS:100 110
CS:100 L 10
CS:100
```

The following example, however, is illegal:

```
CS:100 CS:110
        ^ Error
```

The limit for range is 10000 (hexadecimal). To specify a value of 10000 with only four digits, type 0000 (or 0).

list

A series of *byte* values or *strings*. *list* must be the last parameter on the command line.

Following is an example *list*:

```
fcs:100 42 45 52 54 41
```

string

Any number of characters enclosed in quotation marks. The quotation marks may be either single (’ ’) or double (” “). If the delimiter marks must appear within a string, you must use the double quotation marks. For example, the following strings are legal:

“This ‘string’ is okay.”

“This “”string”” is okay.”

However, this string is illegal:

“This “string” is not okay.”

Note that the double quotation marks are not necessary in the following strings:

“This “string” is not necessary.”

“This ““string”” is not necessary.’

The ASCII values of the characters in the string are used as a list of byte values.

ASSEMBLE

Purpose:

Assembles 8086/8087/8088 mnemonics directly into memory.

Syntax

A[*address*]

Comments

If a syntax error is found, DEBUG responds with the following message, then redisplays the current assembly address:

```
    ^Error
```

All numeric values are hexadecimal and you must type them as 1–4 characters. Also, you must specify prefix mnemonics in front of the opcode to which they refer. You may type them on a separate line, however.

The segment override mnemonics are CS:, DS:, ES:, and SS:. The mnemonic for the far return is RETF. String manipulation mnemonics must explicitly state the string size. For example, use MOVSW to move word strings, and use MOVSB to move byte strings.

The assembler will automatically assemble short, near, or far jumps and calls, depending on byte displacement, to the destination address. You may override these jumps and calls by using a NEAR or FAR prefix, as in the following example:

```
0100:0500   JMP   502        ; a 2-byte short jump
0100:0502   JMP   NEAR 505   ; a 3-byte near jump
0100:505    JMP   FAR 50A    ; a 5-byte far jump
```

You may abbreviate the NEAR prefix to NE, but the FAR prefix cannot be abbreviated.

DEBUG cannot tell whether some operands refer to a word memory location or to a byte memory location. In this case, the data type must be explicitly stated with the prefix, WORD PTR or BYTE PTR. Acceptable abbreviations are WO and BY. For example:

```
NEG    BYTE PTR [128]
DEC    WO [SI]
```

DEBUG also cannot tell whether an operand refers to a memory location or to an immediate operand. So, it uses the common convention that operands enclosed in square brackets refer to memory. For example:

```
MOV      AX,21      ; Load AX with 21H
MOV      AX,[21]    ; Load AX with the
                    ; contents
                    ; of memory location 21H
```

Two popular pseudo-instructions are available with the A (ASSEMBLE) command: the DB opcode, which assembles byte values directly into memory; and the DW opcode, which assembles word values directly into memory. Following are examples of both:

```
DB 1,2,3,4,“THIS IS AN EXAMPLE   ”
DB ‘THIS IS A QUOTATION MARK: ”’
DB “THIS IS A QUOTATION MARK: ’”
DW 1000,2000,3000,“BACH”
```

The A command supports all forms of register indirect commands. For example:

```
ADD     BX,34[BP+2].[SI–1]
POP     [BP+DI]
PUSH    [SI]
```

All opcode synonyms are also supported, as in the next example:

```
LOOPZ  100
LOOPE  100
JA     200
JNBE   200
```

For 8087 opcodes, the WAIT or FWAIT prefixes must be explicitly specified, as in these last examples:

```
FWAIT FADD ST,ST(3)   ; This line assembles
                                  ; an FWAIT prefix
LD TBYTE PTR [BX]     ; This line does not
```

COMPARE

Purpose

Compares the portion of memory specified by range to a portion of the same size beginning at the specified address.

Syntax

Crange address

Comments

If the two areas of memory are identical, there is no display, and DEBUG returns with the MS-DOS prompt. If there are differences, they are displayed in this format:

address1 byte1 byte2 address2

Example

The following commands have the same effect:

```
C100,1FF 300

C100L100 300
```

Each command compares the block of memory from 100 to 1FFH with the block of memory from 300 to 3FFH.

DUMP

Purpose

Displays the contents of the specified range of memory.

Syntax

D[*range*]

Comments

If you specify a *range* of addresses with the D (DUMP) command, the contents of the *range* are displayed. If you don't use parameters with the D command, 128 bytes are displayed at the first address (DS:100) after the address displayed by the previous D command.

The dump is displayed in two portions: a hexadecimal dump (each byte is shown in hexadecimal value) and an ASCII dump (the bytes are shown in ASCII characters). Nonprinting characters are denoted by a period (.) in the ASCII portion of the display. Each display line shows 16 bytes, with a hyphen between the eighth and ninth bytes. At times, displays are split in this section to fit them on the page. Each displayed line begins on a 16–byte boundary.

Examples

If you type the command:

```
dcs:100 110
```

DEBUG displays the dump in the following format:

```
04BA:0100 42 45 52 54 41 ... 4E 44 TOM SAWYER
```

If you simply type the D command, the display is formatted as described above. Each line of the display begins with an address, incremented by 16 from the address on the previous line.

Each subsequent D (typed without parameters) displays the bytes immediately following those last displayed.

If you type the following command, the display is formatted as described above, but 20H bytes are displayed:

```
DCS:100 L 20
```

If you then type the following command, the display is formatted as described above, but all the bytes in the range of lines from 100H to 115H in the CS segment are displayed:

```
DCS:100 115
```

ENTER

Purpose

Enters byte values into memory at the specified *address*.

Syntax

E*address*[*list*]

Comments

If, when using the E (ENTER) command, you type the optional list of values, the byte values are replaced automatically. (If an error occurs, no byte values are changed.)

If you type the address without the optional list, DEBUG displays the address and its contents, repeats the address on the next line, and then waits for your input. At this point, the E (enter) command waits for you to perform one of the following actions:

- Replace a byte value with a value you type. Simply type the value after the current value. If the one you type is not a legal hexadecimal value or if it contains more than two digits, the illegal or extra character is not echoed.
- Press the **space bar** to advance to the next byte. To change the value, simply type the new value as described in the previous action. If, when you press the **space bar**, you move beyond an 8-byte boundary, DEBUG starts a new display line with the address displayed at the beginning.
- Type a hyphen (-) to return to the preceding byte. If you decide to change a byte behind the current position, typing the hyphen returns the current position to the previous byte. When you type the hyphen, a new line is started with its address and byte value displayed.
- Press **Enter** to terminate the E command. The **Enter** key may be pressed at any byte position.

Examples

Suppose you type the following command:

```
ECS:100
```

Now suppose that DEBUG displays the following:

```
04BA:0100   EB._
```

To change this value to, say, 41, type the number **41** at the cursor, as shown:

```
04BA:0100   EB.41_
```

To step through the subsequent bytes, you would press the **space bar** until you saw the following:

```
04BA:0100   EB.41    10.  00.    BC._
```

To change BC to the number 42, for instance, you would type the number at the cursor, as follows:

```
04BA:0100   EB.41    10.  00.    BC.42_
```

Notice that the value 10 should be 6F. To correct this value, you would 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
```

Pressing **Enter** ends the E command and returns you to the DEBUG command level.

FILL

Purpose

Fills the addresses in the specified range with the values in the specified *list*.

Syntax

F*range list*

Comments

If the *range* contains more bytes than the number of values in the *list*, the list will be 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 will occur in all succeeding locations.

Example

Suppose you type the following command:

```
F04BA:100 L 100 42 45 52 54 41
```

DEBUG would now fill memory locations 04BA:100 through 04BA:1FF with the bytes specified. The five values would then be repeated until all the 100H bytes were filled.

GO

Purpose

Executes the program currently in memory.

Syntax

G[=*address* [*addresses*]]

Comments

If you type the G (GO) command by itself, the program currently in memory executes as if it had run outside DEBUG.

If you set =*address*, execution of the G command begins at the address specified. The equal sign (=) is required so that DEBUG can distinguish the start =*address* from the breakpoint *address*es.

With the other optional *address*es set, execution stops at the first *address* encountered, regardless of that address' position in the list of addresses that halt execution or program branching. When program execution reaches a breakpoint, the registers, flags, and decoded instruction are displayed for the last instruction executed. (The result is the same as if you had typed the R (register) command for the breakpoint address.)

You may set up to ten breakpoints, but only at addresses containing the first byte of an 8086 opcode. If you set more than ten breakpoints, DEBUG returns the BP error message.

The user stack pointer must be valid and must have 6 bytes available for this command. The G command uses an IRET instruction to cause a jump to the program under test. The user stack pointer is set, and the user flags, Code Segment register, and Instruction Pointer are pushed on the user stack. (If the user stack is not valid or is too small, the operating system may crash.) An interrupt code (0CCH) is placed at the specified breakpoint address(es).

When DEBUG encounters an instruction with the breakpoint code, it restores all breakpoint addresses to their original instructions. If you don't halt execution at one of the breakpoints, the interrupt codes are not replaced with the original instructions.

Example

Suppose you type the following command:

```
    GCS:7550
```

The program currently in memory would execute up to the address 7550 in the CS segment. DEBUG would then display registers and flags, after which the G command would terminate.

After DEBUG has encountered a breakpoint, if you type the G command again the program runs as if you had typed the filename at the MS-DOS command level. The only difference is that program execution begins at the instruction after the breakpoint, rather than at the usual start address.

HEX

Purpose

Performs hexadecimal arithmetic on the two specified parameters.

Syntax

H*value value*

Comments

First, DEBUG adds the two parameters, then subtracts the second parameter from the first. The results of these actions are displayed on one line--first the sum, then the difference.

Example

Suppose you type the following command:

```
H19F 10A
```

In response, DEBUG would perform the calculations and then display the following result:

```
02A9 0095
```

INPUT

Purpose

Inputs and displays one byte from the port specified by *value*.

Syntax

I*value*

Comment

A 16-bit port address is allowed.

Example

Suppose you type the following command:

```
I2F8
```

Suppose also that the byte at the port is 42H. DEBUG would input the byte and then display the following:

```
42
```

LOAD

Purpose

Loads a file into memory.

Syntax

L[*address* [*drive: record record*]]

Comments

Set BX:CX to the number of bytes read. The file must have been named either when you started DEBUG or with the N (NAME) command. Both the DEBUG invocation and the N command format a filename properly in the normal format of a File Control Block at CS:5C.

If you use the L (load) command without any parameters, DEBUG loads the file into memory beginning at address CS:100 and sets BX:CX to the number of bytes loaded. If you type the L command with an address parameter, loading begins at the memory location specified by the *address*. If you use the L command with all parameters included, absolute disk sectors are loaded, instead of a file.

Each *record* is taken from the specified drive: (the drive designation is numeric here: 0=A:, 1=B:, 2=C:, etc.). DEBUG begins loading with the first specified *record*, and continues until the number of sectors in the second *record* have been loaded.

Example

Suppose once you have started DEBUG that you type the following command:

```
-NFILE.COM
```

Now, to load FILE.COM, you would simply type the L command. DEBUG would then load the file and display the DEBUG prompt. Suppose now that you want to load only portions of a file or certain records from a disk. To do this, you would type the following:

```
L04BA:100 2 0F 6D
```

DEBUG would then load 109 (6D hex) records, beginning with logical record number 15, into memory beginning at address 04BA:0100. Then, once it had loaded the records, DEBUG would simply return the hyphen (–) prompt.

If the file has an .EXE extension, it would be relocated to the load address specified in the header of the .EXE file. The address parameter is always ignored for .EXE files. The header itself is stripped off the .EXE file before it is loaded into memory. So, the size of an .EXE file on disk will differ from its size in memory.

If the file was named by the N (NAME) command, or specified when you started DEBUG, as a .HEX file, then typing the L command with no parameters would cause DEBUG to load the file beginning at the address specified in the .HEX file. If the L command included the option *address*, DEBUG would add the address specified in the L command to the *address* found in the .HEX file to determine the start address at which to load the file.

MOVE

Purpose

Moves the block of memory specified by *range* to the location beginning at the specified *address*.

Syntax

M*range address*

Comments

Overlapping moves (i.e., moves where part of the block overlaps some of the current addresses) are always performed without loss of data. Addresses that could be overwritten are moved first. For moves from higher to lower addresses, the sequence of events is to first move the data beginning at the block's lowest address and then work toward the highest. For moves from lower to higher addresses, the sequence is to first move the data beginning at the block's highest address and then work toward the lowest.

Note that if the addresses in the block being moved will not have new data written to them, the data in the block before the move will remain. The M (MOVE) command copies the data from one area into another, in the sequence described, and writes over the new addresses. This action is why the sequence of the move is important.

Example

Suppose you type the following command:

```
MCS:100 110 CS:500
```

In response, DEBUG would first move address CS:110 to CS:510, then move CS:10F to CS:50F, and so on until CS:100 is moved to CS:500. To review the results of the move, you could type the D command, using the same *address* as you used with the M command.

NAME

Purpose

Sets filenames.

Syntax

N*filename* [*filename...*]

Comments

The N (NAME) command performs two functions. First, it assigns a filename for a later L (LOAD) or W (WRITE) command. So, if you start DEBUG without naming a file to be debugged, you must type the command N*filename* before a file can be loaded. Second, the N command assigns filename parameters to the file being debugged. In this case, N accepts a list of parameters used by the file being debugged.

Note that these two functions overlap. Consider, for example, the following set of DEBUG commands:

```
-NFILE1.EXE
-L
-G
```

The N command would use these commands to perform the following steps:

1. It would first assign the filename FILE1.EXE to the file to be used in any later L or W commands.
2. It would also assign the FILE1.EXE filename to the first filename parameter used by any program that is later debugged.
3. The L command would then load FILE1.EXE into memory.
4. The G (GO) command would cause FILE1.EXE to be run with FILE1.EXE as the single filename parameter (that is, FILE1.EXE would be run as if FILE1.EXE had been typed at the command level).

A more useful chain of commands might look like this:

```
-NFILE1.EXE
-L
-NFILE2.DAT FILE3.DAT
-G
```

In this example, the N command sets FILE1.EXE as the filename for the subsequent L command, which loads FILE1.EXE into memory. The N command is then used again, this time to specify the parameters to be used by FILE1.EXE. Finally, when the G command is run, FILE1.EXE is executed as if FILE1 FILE2.DAT FILE3.DAT had been typed at the MS–DOS command level.

Note that if you were to execute a W command now, then FILE1.EXE--the file being debugged--would be saved with the name FILE2.DAT. To avoid this kind of result, you should always execute an N command before either an L or W command.

There are four regions of memory that can be affected by the N command:

CS:5C	FCB for file 1
CS:6C	FCB for file 2
CS:80	Count of characters
CS:81	All characters typed

The first filename parameter that you specify for the N command has a file control block (FCB) set up at CS:5C. If you name a second filename parameter, an FCB is set up for this parameter beginning at CS:6C.

The number of characters typed in the N command (exclusive of the first character, N) is given at location CS:80. The actual stream of characters given by the N command (again, exclusive of the letter N) begins at CS:81. Note that this stream of characters may contain switches and delimiters that would be legal in any command typed at the MS–DOS command level.

Example

A typical use of the N command is as follows:

```
DEBUG PROG.COM
-NPARAM1 PARAM2/C
-G
```

In this case, the G command executes the file in memory as if you had typed the following command line:

```
PROG PARAM1 PARAM2/C
```

Testing and debugging therefore reflect a normal runtime environment for PROG.COM.

OUTPUT

Purpose

Sends the specified byte to the output port specified by value.

Syntax

Ovalue byte

Comment

A 16-bit port address is allowed.

Example

Suppose you want to DEBUG to output the byte value 4F to output port 2F8. To do this you could simply type the following command:

```
O2F8 4F
```

QUIT

Purpose

Terminates the DEBUG utility.

Syntax

Q

Comments

The Q (QUIT) command takes no parameters and exits DEBUG without saving the file you're currently working with. You are returned to the MS–DOS command level.

Example

To end the debugging session, simply type the following and press the **Enter** key:

```
    Q
```

DEBUG terminates, and control returns to the MS–DOS command level.

REGISTER

Purpose

Displays the contents of one or more CPU registers.

Syntax

R[*register–name*]

Comments

If you do not type a *register–name*, the R (REGISTER) command dumps the register storage area and displays the contents of all registers and flags.

If you do type a *register–name*, the 16–bit value of that register is displayed in hexadecimal, and a colon then appears as a prompt. You can now either type a value to change the register, or press the **Enter** key if you don't want a change.

Following is a list of the valid *register–names*:

AX
BX
CX
DX
SP

BP
SI
DI
DS
ES

SS
CS
IP
PC
F

(IP and PC both refer to the Instruction Pointer.)

Any other entry for *register–name* results in a BR error message.

If you type F as the *register–name*, DEBUG displays each flag with a two–character alphabetic code. To change any flag, type the opposite two–letter code. The flags are then either set or cleared.

The flags are listed below with their codes for SET and CLEAR:

FLAG NAME	SET	CLEAR
Overflow	OV	NV
Direction	DN Decrement	UP Increment
Interrupt	EI Enabled	DI Disabled
Sign	NG Negative	PL Plus
Zero	ZR	NZ
Auxiliary Carry	AC	NA
Parity	PE Even	PO Odd
Carry	CY	NC

Whenever you type the RF command, the flags are displayed (in the order shown in the previous table) in a row at the beginning of a line. At the end of the list of flags, DEBUG displays a hyphen (–).

You may enter new flag values in any order as alphabetic pairs. You do not have to leave spaces between these values. To exit the R command, press the **Enter** key. Any flags for which you did not specify new values remain unchanged.

If you type more than one value for a flag, DEBUG returns a DF error message. If you enter a flag code other than one of those shown in the table above, DEBUG returns a BF error message. In both cases, the flags up to the error in the list are changed; those flags at and after the error are not.

When you start DEBUG, the segment registers are set to the bottom of free memory, the Instruction Pointer is set to 0100H, all flags are cleared, and the remaining registers are set to zero.

Examples

If you type the following command, DEBUG displays all registers, flags, and the decoded instruction for the current location:

```
R
```

If the location is CS:11A, for example, the display will look similar to this:

```
AX=0E00  BX=00FF  CX=0007  DX=01FF  SP=039D  BP=0000
SI=005C  DI=0000  DS=04BA  ES=04BA  SS=04BA  CS=O4BA
P=011A   NV UP DI NG NZ AC PE NC
04BA:011A   CD21            INT       21
```

If you then type the following command, DEBUG will display these flags:

```
RF
NV UP DI NG NZ AC PE NC - _
```

Now, you could type any valid flag designation, in any order, with or without spaces. For example:

```
NV UP DI NG NZ AC PE NC - PLEICY
```

In response, DEBUG would display the DEBUG prompt. To see the changes, type either the R or RF command:

```
RF
NV UP EI PL NZ AC PE CY - _
```

Press the **Enter** key to leave the flags this way or to specify different flag values.

SEARCH

Purpose

Searches the specified range for the specified list of bytes.

Syntax

Srange list

Comments

The *list* may contain one or more bytes, each separated by a space or comma. If the *list* contains more than one byte, only the first address of the byte string is returned. If the *list* contains only one byte, all addresses of the byte in the *range* are displayed.

Example

Suppose you type the following command:

```
SCS:100 110 41
```

DEBUG would display a response similar to this:

```
04BA:0104
04BA:010D
–type:
```

TRACE

Purpose

Executes one instruction and displays the contents of all registers, flags, and the decoded instruction.

Syntax

T[=*address*] [*value*]

Comments

If you include the =*address* option in the T (TRACE) command, tracing occurs at the specified =*address*. The *value* option causes DEBUG to execute and trace the number of steps specified by *value*.

The T command uses the hardware trace mode of the 8086 or 8088 microprocessor. Consequently, you may also trace instructions stored in ROM (Read Only Memory).

Example

Suppose you type the following command:

```
T
```

In response, DEBUG would return a display of the registers, flags, and decoded instruction for that one instruction. Assuming, for this example, that the current position is 04BA:011A, DEBUG might return the following display:

```
AX=0E00  BX=00FF  CX=0007  DX=01FF  SP=039D  BP=0000
SI=005C  DI=0000  S=04BA   ES=04BA  SS=04BA  CS=O4BA
IP=011A  NV UP DI NG NZ AC PE NC
04BA:011A   CD21           INT      21
```

If you type the following command, DEBUG executes sixteen (10 hex) instructions beginning at 011A in the current segment, and then displays all registers and flags for each instruction as it is executed. The display scrolls away until the last instruction is executed, and then stops. Now you can see the register and flag values for the last few instructions performed:

```
T=011A 10
```

Remember that if you want to study the registers and flags for any instruction (at any time), you can press **Ctrl–S** to stop the display from scrolling.

UNASSEMBLE

Purpose

Disassembles bytes and displays the source statements that correspond to them, with addresses and byte values.

Syntax

U[*range*]

Comments

The display of disassembled code looks like a listing for an assembled file. If you type the U (UNASSEMBLE) command without parameters, 20 hexadecimal bytes are disassembled at the first address after that displayed by the previous U command. If you type the U command including the *range* parameter, then DEBUG disassembles all bytes in *range*. But if you specify *range* only as an *address*, then 20H bytes are disassembled.

Example

Suppose you type the following command:

```
U04BA:100 L10
```

In response, DEBUG would disassemble 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
```

Now, suppose you type the following:

```
U04ba:0100 0108
```

The display would now show:

```
04BA:0100   206472   AND   [SI+72],AH
04BA:0103   69       DB    69
04BA:0104   7665     JBE   016B
04BA:0106   207370   AND   [BP+DI+70],DH
```

If the bytes in some addresses are altered, the disassembler alters the instruction statements. You can then type the U command for the changed locations, the new instructions viewed, and the disassembled code used to edit the source file.

WRITE

Purpose

Writes the file being debugged to a disk file.

Syntax

W[*address* [*drive: record record*]]

Comments

If you do not use parameters with the W (WRITE) command, BX:CX must already be set to the number of bytes to be written; the file is written beginning from CS:100. If you type the W command with just an address, then the file is written beginning at that address. If you have used a G (GO) or T (TRACE) command, you must reset BX:CX before using the W command without parameters.

Note that if a file is loaded and modified, the name, length, and starting address are all set correctly to save the modified file (as long as the length has not changed).

You must have named the file either with the initial DEBUG startup command or with the N (name) command (refer to the N command earlier in this manual). Both the DEBUG startup command and the N command properly format a filename in the normal format of a File Control Block at CS:5C.

If you include parameters when you use the W command, the write begins from the memory address specified; the file is written to the specified drive: (the drive name is numeric here--0=A:, 1=B:, 2=C:, etc.). DEBUG writes the file beginning at the logical record number specified by the first record. DEBUG then continues to write the file until the number of sectors specified in the second record have been written.

Writing to absolute sectors is extremely risky because the process bypasses the file handler.

Examples

If you type the following command, DEBUG will write the file to disk and then display the DEBUG prompt:

```
W
```

Two examples are shown below.

```
W
-
WCS:100 1 37 2B
```

DEBUG writes out the contents of memory to the disk in drive B:, beginning with the address CS:100. The data written out starts in the disk logical record number 37H and consists of 2BH records. When the write is complete, DEBUG displays the following prompt:

```
WCS:100 1 37 2B
- _
```

DEBUG ERROR MESSAGES

During a DEBUG session, you may receive any of the following error messages. Each error ends the DEBUG command under which it occurred, but does not end DEBUG itself.

ERROR CODE DEFINITION

BF

Bad flag
You attempted to change a flag, but the characters you typed were not one of the acceptable pairs of flag values. See the R (REGISTER) command for the list of acceptable flag entries.

BP

Too many breakpoints
You specified more than ten breakpoints as parameters to the G (GO) command. Retype the G command using ten or fewer breakpoints.

BR

Bad register
While using the R command you typed an invalid register name. See the R command for the list of valid register names.

DF

Double flag
You typed two values for one flag. You may specify a flag value only once per RF command. flag value only once per RF command.

Appendix A

The System Setup Utility

INTRODUCTION

This section describes the System Setup utility, including

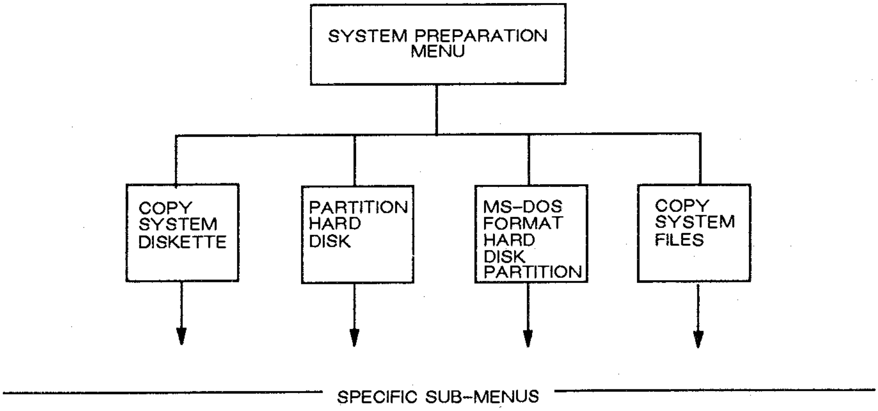
- how to use the Setup utility
- descriptions and System Preparation procedures.

General Description of the Setup Utility

There are four distribution diskettes provided with your PowerMate personal computer. They contain

- the MS–DOS system software used to start the PowerMate
- additional MS–DOS utilities
- the System Setup diskette containing utility software that allows you to prepare your system for use.

The Setup utility consists of a tree–like organization of menus and submenus. As you make selections and progress downward through the menus, you go from general categories of operation to more specific operations. Figure A–1 illustrates the overall organization of the menu structure.



*NOTE: THE SYSTEM PREPARATION MENU CAN BE ACCESSED ONLY IF NO DISCREPANCIES ARE DETECTED DURING THE POWER-ON SELF-TEST.

Figure A-1 The Menu Structure of the Setup Utility

How the Setup Utility Works

When you first turn on your PowerMate, the system automatically identifies its hardware components. For example, it determines how much memory is present, whether the system clock/calendar is set, and which communication ports and display adapters are installed. This process is called the Power-On Self-Test.

The system compares the hardware it finds with the setup information stored in its permanent memory (CMOS). Any discrepancies must be corrected before you can operate your PowerMate.

If a discrepancy is detected, you must run the Hardware System Setup provided with your PowerMate to correct the information in CMOS.

NOTE

> If no discrepancies are detected, the system does not automatically run the Setup program. Instead, the system prompt (A>) is displayed after you start up your PowerMate. From here, you can perform any operation you desire, including the rest of the Setup program. Refer to the next section, "USING THE SETUP UTILITY," for more information.

When the system definition is completed to your satisfaction, the PowerMate reboots (starts up again) and performs a Power–On Self–test. If no new discrepancies are found, you can use the Setup software to perform other necessary operating system setup operations, such as

- Making an exact copy of the MS–DOS System distribution diskettes.
- Making a "simplified" MS–DOS System diskette.

If you have one or more hard disk drive(s) in your PowerMate, you can

1. Divide the disk(s) into MS–DOS partitions.
2. Format MS–DOS partition(s) and place the MS–DOS system files into it/them.
3. Place a volume name on the partition(s).
4. Create a simple subdirectory structure for utilities, drivers, and so on.
5. Prepare the partition to automatically install various device drivers at boot time.
6. Copy utility programs and files from the distribution diskette into the MS–DOS partition(s).

These setup procedures are explained in detail in "THE SYSTEM PREPARATION MENU."

USING THE SETUP UTILITY

To use the Setup utility, follow these steps:

1. Make sure all options (such as disk drives, video controller boards, and math co–processors) have been installed, and that all hard disk bad track information lists have been retrieved from inside the system unit and are at hand.
2. Boot (start up) your PowerMate using the Hardware System Setup diskette. (Refer to the PowerMate owner's guide for information on how to start up your system and load a diskette.)

   The system performs a Power–On Self–Test to check the accuracy of setup information stored in permanent memory in the PowerMate.

3. If discrepancies are found during the Power–On Self–Test, the following message is displayed:

   Invalid configuration information – please run SETUP program.
   Strike F1 key to continue

   Insert the Hardware System Setup diskette in drive A:
   Press the **F1** key and the system will continue to boot.

   After the system boots, it automatically starts the Hardware System Setup program. Refer to the Hardware System Setup guide for information on how to fix discrepancies found during the booting process.

   If no discrepancies are found during start–up, or after they have been corrected with the Hardware System Setup, boot the Power-Mate with the MS–DOS System Setup diskette.

4. Run the Setup utility by typing

   SETUP

   and pressing **Enter**. The first menu of the Setup utility, the System Preparation menu, is displayed.

In each of the menus, the available options are listed near the middle of the screen, with the topmost selection highlighted. To make a selection, perform the following steps:

1. Press the up or down arrow key to move the highlighting cursor to another choice.
2. When the highlighting cursor is positioned on the desired selection, press **Enter**.

The Setup utility will proceed with the operation you selected.

As specific choices are highlighted in each menu, a line of descriptive text appears near the bottom of the screen. This text describes the specific action that will occur for the highlighted selection.

Optionally, you can return from each submenu to the preceding menu. When you return to a given menu, the operation you most recently performed on that menu is highlighted.

You can also return to the previous menu by pressing **Esc**. If you press **Esc** at the System Preparation menu, a prompt appears asking you if you want to leave the Setup program. If you answer Yes, the system exits from the Setup program. If you answer No, you remain in the Setup program and you can continue with other setup operations.

THE SYSTEM PREPARATION MENU

The System Preparation menu appears on your screen when you begin using the Setup utility. You will return to this menu after completing the System Preparation functions. You must return to the System Preparation menu to exit from the Setup program.

The appearance of the System Preparation menu will vary depending on whether any setup discrepancies were found during the Power-On Self-Test. If discrepancies were found, you will be asked to exit from the Setup program and reboot with the Hardware System Setup diskette.

This section describes the System Preparation menu and tells you how to

- use the System Preparation menu
- make copies of the MS-DOS System distribution diskettes.

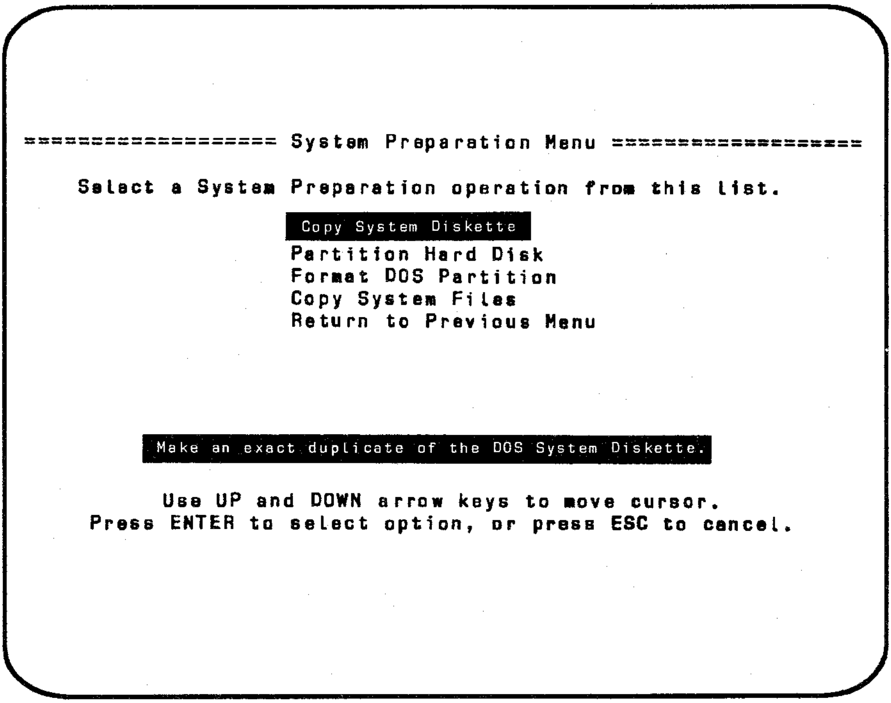
If your PowerMate has one or more hard disk drives installed, you can also

- divide a hard disk into MS-DOS partitions
- format an MS-DOS partition
- copy system files to an MS-DOS partition.

Using the System Preparation Menu

To use the System Preparation menu, your PowerMate must be able to access disk drives. For this reason, the System Preparation functions can be used only after you have successfully completed the Hardware System Setup operations.

A screen similar to the following appears.



```
=================== System Preparation Menu ==================

     Select a System Preparation operation from this list.

                        Copy System Diskette
                        Partition Hard Disk
                        Format DOS Partition
                        Copy System Files
                        Return to Previous Menu



              Make an exact duplicate of the DOS System Diskette.

               Use UP and DOWN arrow keys to move cursor.
          Press ENTER to select option, or press ESC to cancel.
```

Initially, the cursor is positioned on the topmost selection. You can move the cursor to another selection by pressing the up or down arrow keys. When the cursor is on the desired selection, press **Enter**.

To return to the Entry menu, press **Esc**.

How to Copy the MS-DOS System Diskette

Use the MS-DOS System distribution diskettes you received with your PowerMate as "master" diskette from which "working" copies are made. After you have copied the master MS-DOS System distribution diskettes, store them in a safe place.

Using the Copy DOS System Diskette option, you can

- make a duplicate copy of the MS-DOS System distribution diskettes you received with your PowerMate
- make an MS-DOS System diskette that includes only a selected subset of all MS-DOS commands and utilities. This copy process saves space and greatly reduces the number of files on the diskette. This type of diskette is referred to as the "simplified" DOS System diskette.

It is strongly recommended that you make at least one duplicate copy of the MS-DOS System diskette.

To make a duplicate of the MS-DOS System diskette you received with your PowerMate, follow the instructions below. If you want to make a simplified MS-DOS System diskette, follow the instructions in "HOW TO MAKE A SIMPLIFIED DOS SYSTEM DISKETTE."

HOW TO MAKE DUPLICATE COPIES

To make duplicate copies of the master MS-DOS System distribution diskettes, you need

- the MS-DOS System distribution diskettes you received with your PowerMate
- one blank diskette for each working copy of a master MS-DOS System distribution diskette. The blank diskette(s) must be of the same type as the master system distribution diskette.

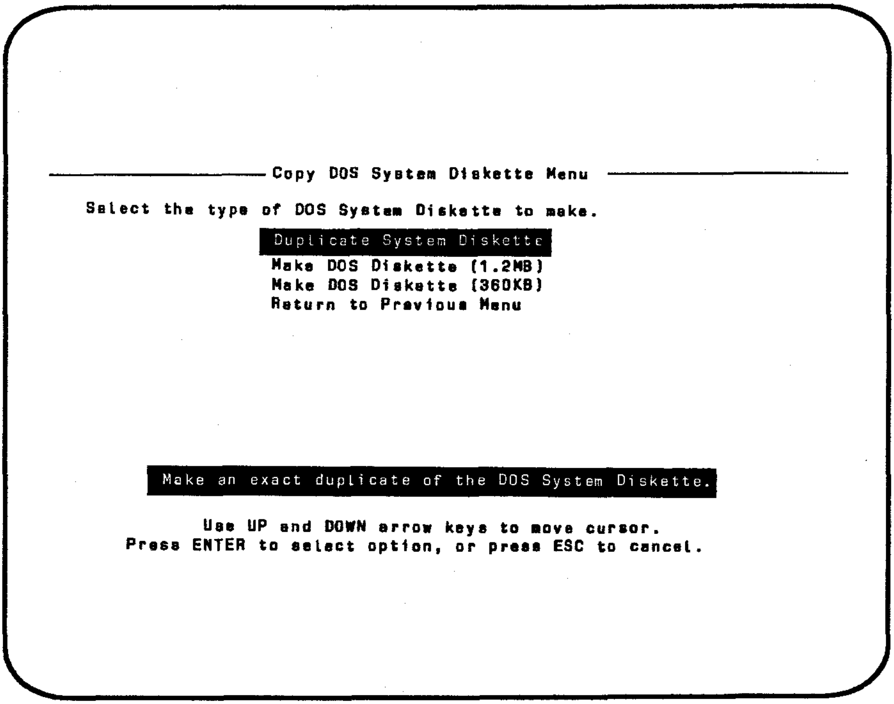
Perform the following steps to make the duplicate copy:

1. Make sure the MS-DOS System distribution diskettes are write protected.

   A 5 1/4-inch diskette is write-protected when it has no notch or when the notch is covered by a tab.

   A micro-diskette is write-protected when the write-protect tab is slid to expose the hole.

2. Select the Copy DOS System Diskette option from the System Preparation Menu. A screen similar to the following is displayed.



```
———————————— Copy DOS System Diskette Menu ————————————
   Select the type of DOS System Diskette to make.
                    Duplicate System Diskette
                    Make DOS Diskette (1.2MB)
                    Make DOS Diskette (360KB)
                    Return to Previous Menu



       Make an exact duplicate of the DOS System Diskette.

              Use UP and DOWN arrow keys to move cursor.
       Press ENTER to select option, or press ESC to cancel.
```

3. Initially, the cursor highlights the Duplicate System Diskette option. Press **Enter** when Duplicate System Diskette is highlighted.

   The screen is cleared and the following message is displayed:

   Insert SOURCE diskette in drive A:
   Press any key when ready ...

4. If the MS-DOS System diskette is not already in drive A, insert it. This is the "source" diskette.

5. Press any key (e.g., the space bar) to continue.

   A message similar to the following is displayed:

   ```
   Copying 80 Tracks
   15 Sectors/Track, 2 Sides
   ```

   Next, the following message is displayed:

   ```
   Insert TARGET diskette in drive A:
   Press any key when ready ...
   ```

6. The blank diskette, which will become your working copy of the master MS-DOS System diskette, is the "target" diskette. Insert this diskette into drive A.

7. Press any key to continue. If the copy process is not yet complete, the following message is displayed:

   ```
   Insert SOURCE diskette in drive A:
   Press any key when ready...
   ```

8. Reinsert the source diskette and press any key. The following message is displayed:

   ```
   Insert TARGET diskette in drive A:
   Press any key when ready...
   ```

9. Reinsert the target diskette and press any key to continue the copy process.

   You will alternately be asked to insert the source and target diskettes until the copy process is complete.

   When the copy operation is completed, the following message is displayed:

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

10. Press **Y** or **y** to make a copy of the next MS-DOS System distribution diskette. Press **N** or **n** to return to the System Preparation menu.

11. Store your master MS-DOS System diskette in a safe place. Label your working copy of the MS-DOS System diskette and then reinsert it into drive A. At this time you can continue with the System Preparation process.

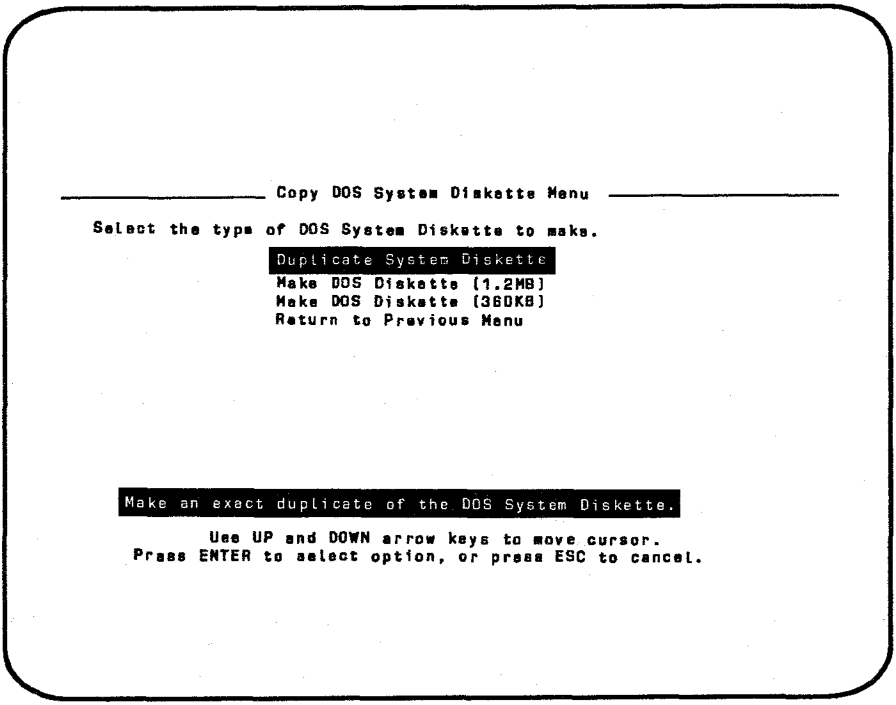
HOW TO MAKE A SIMPLIFIED DOS SYSTEM DISKETTE

To make a simplified MS-DOS System diskette, you need

- the MS-DOS System diskette you received with your PowerMate
- one 1.2 MB (high-capacity), 360 KB (double-density), 720 KB, or 1.44 MB (3 1/2-inch) diskette for each MS-DOS System diskette you intend to make.

To make a simplified MS-DOS System diskette, perform these steps:

1. Make sure the MS-DOS System diskette is write protected.
2. Select the Copy DOS System Diskette option from the System Preparation menu. A screen similar to the following is displayed.



```
 ________________ Copy DOS System Diskette Menu ________________

   Select the type of DOS System Diskette to make.
                  Duplicate System Diskette
                  Make DOS Diskette (1.2MB)
                  Make DOS Diskette (360KB)
                  Return to Previous Menu


       Make an exact duplicate of the DOS System Diskette.

             Use UP and DOWN arrow keys to move cursor.
       Press ENTER to select option, or press ESC to cancel.
```

3. Initially, the cursor highlights Duplicate System Diskette. The format options displayed on the menu are based on your hardware system setup. Use the up arrow and down arrow keys on the numeric keypad to move the cursor among the various menu items.

   To make a 1.2 MB DOS System diskette, move the cursor to Make DOS System Diskette (1.2 MB) and press **Enter**.

   To make a 360 KB DOS System diskette, move the cursor to the Make DOS System Diskette (360 KB) option and press **Enter**.

   To make a 720 KB DOS System diskette, move the cursor to the Make DOS System Diskette (720 KB) option and press **Enter**.

   To make a 1.4 MB DOS System diskette, move the cursor to Make DOS System Diskette (1.2 MB) and press **Enter**.

4. At this time the screen clears and a message similar to the following is displayed:

   ```
   Insert new diskette for drive A:
   and strike ENTER when ready
   ```

   Remove the master MS-DOS System Setup diskette. Insert the diskette you are preparing into the indicated drive and press **Enter**.

5. While the diskette is being formatted the message

   ```
   Formatting...Format complete
   System transferred

   Format another (Y/N)?
   ```

   is displayed.

6. Press **N** or **n** in response to the prompt to format another diskette. Certain MS-DOS system utilities and files will be copied to the diskette. When the entire operation is completed, the Copy DOS System Diskette menu is displayed.

7. At this time, you can

   - perform another selection under the Copy DOS System Diskette menu
   - select Return to a Previous Menu to return to the System Preparation menu.

Overview of Hard Disk Preparation

To prepare a hard disk on the PowerMate, the following options are available from the System Preparation menu:

- Partition hard disk -- Creates one or more MS-DOS partitions. You can make a partition the "active partition" if you want to make it bootable.
- Format DOS partition -- Formats the hard disk partition for MS-DOS. You can copy the MS-DOS system to the active partition if the partition is to be bootable.
- Copy system files -- Copies MS-DOS utilities and system files to the partition.

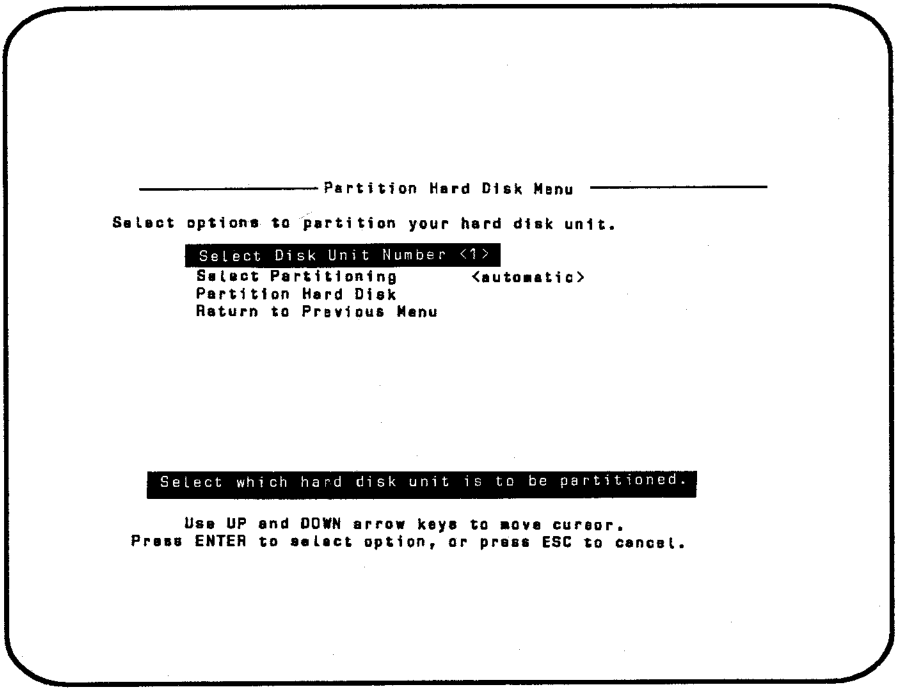
Each of these options is described in detail in the following subsections.

Partition Hard Disk

Partition Hard Disk divides a hard disk into separately accessible data areas called partitions.

When created, a partition is assigned a type according to the operating system that will be used to access it. The Partition Hard Disk option creates MS-DOS partitions.

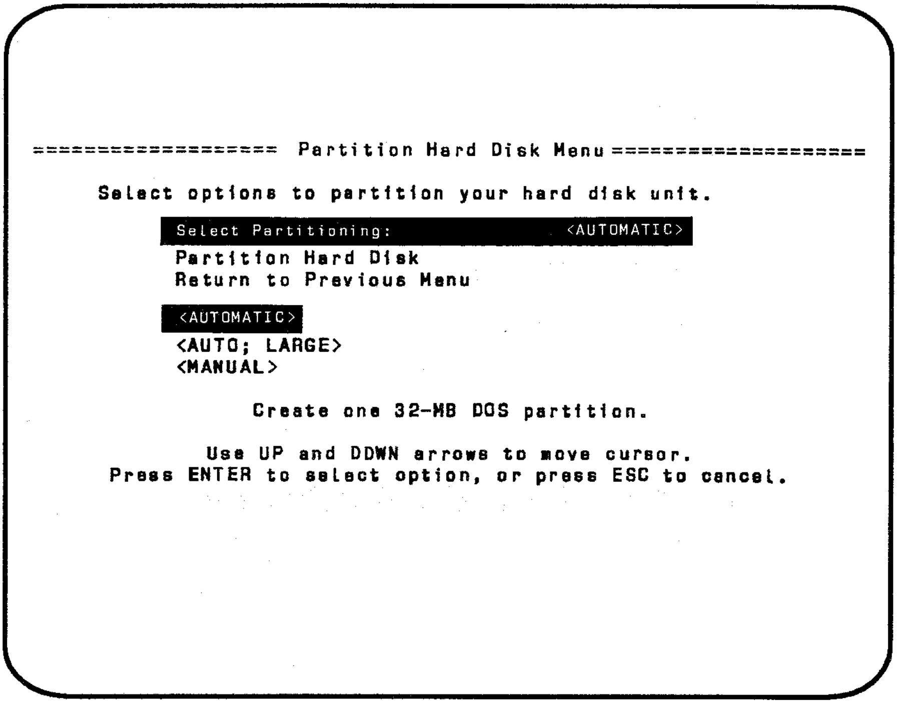
1. Select Partition Hard Disk from the System Preparation menu. The following screen is displayed.



```
                    Partition Hard Disk Menu

Select options to partition your hard disk unit.

         Select Disk Unit Number <1>
         Select Partitioning         <automatic>
         Partition Hard Disk
         Return to Previous Menu



    Select which hard disk unit is to be partitioned.

           Use UP and DOWN arrow keys to move cursor.
   Press ENTER to select option, or press ESC to cancel.
```

2. The first item on the menu, Select Disk Unit Number, appears only if your PowerMate has more than one hard disk drive. In this case, choose the hard disk drive that you want to partition: Disk Unit 1 for the first disk unit, or Disk Unit 2 for the second disk unit.

3. Position the cursor on Select Partitioning and press **Enter**. A screen similar to the following is displayed.



```
=================== Partition Hard Disk Menu ====================

    Select options to partition your hard disk unit.

        Select Partitioning:                   <AUTOMATIC>
        Partition Hard Disk
        Return to Previous Menu

        <AUTOMATIC>
        <AUTO; LARGE>
        <MANUAL>

              Create one 32-MB DOS partition.

            Use UP and DOWN arrows to move cursor.
     Press ENTER to select option, or press ESC to cancel.
```

You can choose to partition the hard disk automatically or manually.

If you have one or more NEC 20–MB, 40–MB, 42–MB, 66–MB, 80–MB, 130–MB, 140–MB, or 300–MB hard disk drives installed in your PowerMate, you can perform automatic partitioning of the drive(s). Automatic creates one 32–MB partition, if possible.

Automatic large partition creates one partition whose size is the full size of the hard drive (for example, one 66–MB partition).

Manual partitioning invokes the FDISK utility, allowing you to create partitions to suit your needs. If you manually partition the hard disk, you must specify partition size, and activation status.

The first partition of Disk Unit 1 is assigned drive letter C, and subsequent partitions (on Disk Unit 1 if applicable, or continuing to Disk Unit 2) are assigned drive letters D, E, F, and so on.

4. If you decide not to partition the disk at this time, select Exit Partition Hard Disk or press **Esc**.

5. After you choose how you want the hard disk to be partitioned, select Partition Hard Disk to begin the partitioning process.

   If the disk is already partitioned, you are warned of this and asked to confirm your decision to partition the disk.

   If you chose automatic partitioning, you will be returned to the System Preparation menu when partitioning is complete.

   If you chose to perform manual partitioning, control is transferred to the MS–DOS utility called FDISK. To use FDISK, follow the instructions provided in Appendix H.

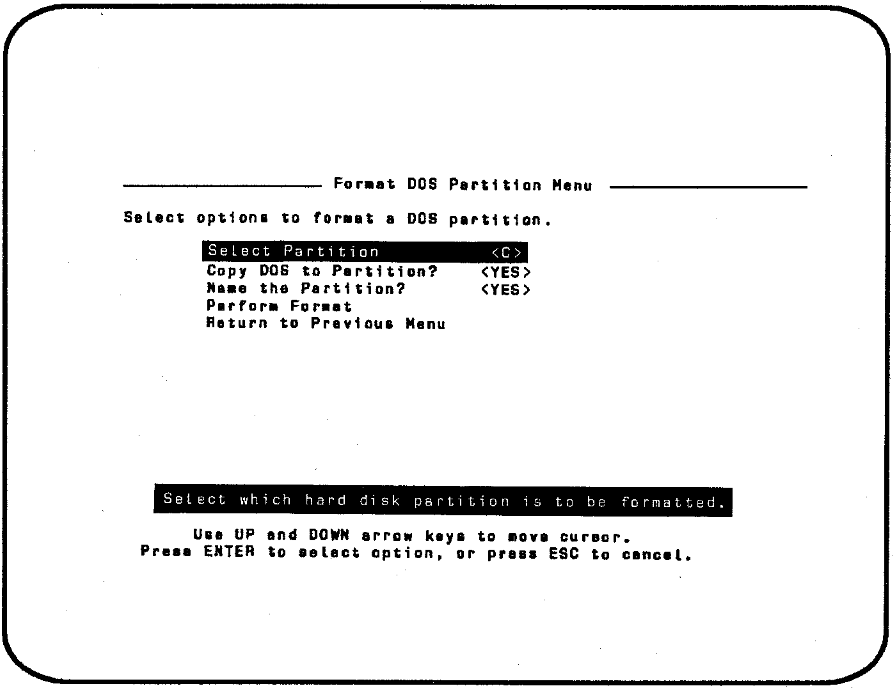
Format DOS Partition

Format DOS Partition performs the following tasks:

- Initializes the selected hard disk partition to a format that is compatible with MS–DOS.
- Checks the partition for any tracks that were flagged as defective tracks. In this way, MS–DOS can avoid using these tracks.
- Initializes the root directory and File Allocation Table and writes the bootstrap system loader to the disk.
- (Optional) Copies special MS–DOS system files to the formatted partition. A partition that contains these MS–DOS system files and which is also the active partition can be used to boot MS–DOS on the PowerMate.
- (Optional) Lets you assign a volume name to the formatted partition.

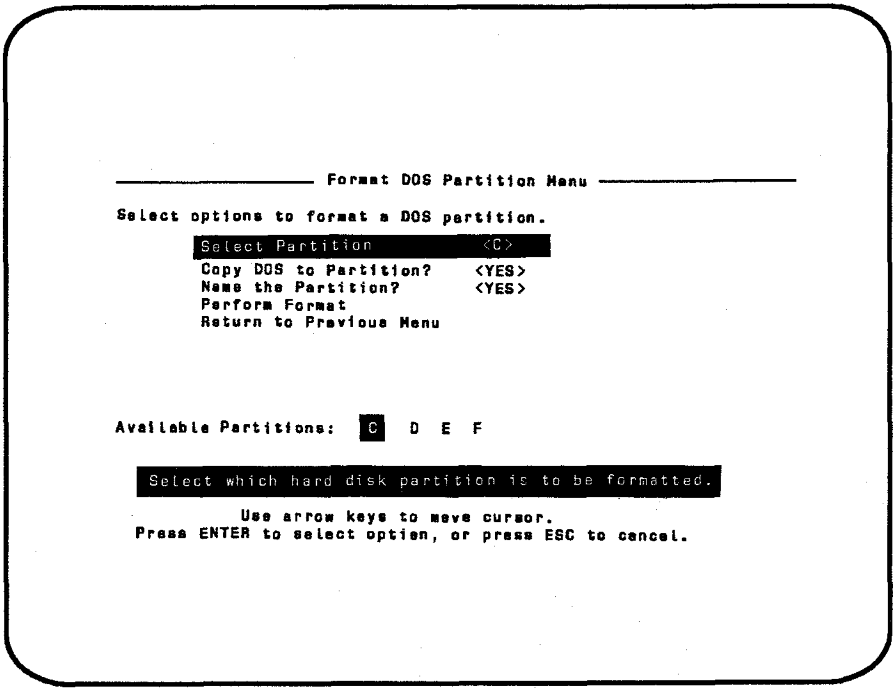
To format a DOS partition, perform the following steps:

1. Select Format DOS Partition from the System Preparation menu. A screen similar to the following is displayed.



```
———————————— Format DOS Partition Menu ————————————
Select options to format a DOS partition.

        Select Partition                <C>
        Copy DOS to Partition?          <YES>
        Name the Partition?             <YES>
        Perform Format
        Return to Previous Menu


  Select which hard disk partition is to be formatted.

        Use UP and DOWN arrow keys to move cursor.
  Press ENTER to select option, or press ESC to cancel.
```

2. Initially, the cursor highlights Select Partition. To specify the partition to be formatted, press **Enter** when Select Partition is highlighted. The cursor moves to the lower portion of the screen, then the available partitions are displayed (similar to the following).



```
              ───────────── Format DOS Partition Menu ─────────────

Select options to format a DOS partition.
         Select Partition               <C>
         Copy DOS to Partition?         <YES>
         Name the Partition?            <YES>
         Perform Format
         Return to Previous Menu


Available Partitions:   C   D   E   F

   Select which hard disk partition is to be formatted.

                 Use arrow keys to move cursor.
     Press ENTER to select option, or press ESC to cancel.
```

To select a partition, move the cursor to the desired partition and press **Enter**. The cursor returns to the Select Partition option with the newly-selected partition displayed.

3. If you specify YES for the Copy DOS to Partition? option, the MS-DOS system files will be copied to the partition. This is necessary if you want the partition to be bootable.

4. If you specify YES for the Name the Partition? option, you will be prompted to enter the volume name after the partition is formatted. A volume name is an internal name, up to 11 characters long, that will help you identify the partition.

5. When the format parameters are set correctly, select the Perform Format option to begin the formatting process.

   The screen clears. The MS-DOS FORMAT utility is invoked, and it uses the parameters you have just specified.

   If the partition was previously formatted, the following message is displayed:

   ```
   Enter current Volume Label for drive C:
   ```

   To continue the format process, enter the volume label for drive C:.

6. The following message is then displayed:

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

   If you do *not* want to lose the data already on the disk, type **N** or **n** and press **Enter** to cancel the format process.

   If you want to format the partition, type **Y** or **y** and press **Enter** to begin the format process.

   While the partition is being formatted the head and cylinder currently being formatted are displayed.

   ```
   Head: 0    Cylinder: 0
   ```

   When the format operation finishes, the message

   ```
   Format  complete
   ```

   is displayed.

7. If you previously specified YES for Name the Partition?, you are prompted to enter a volume name.

   If you want to include a volume name, type the name and press **Enter**.

   If you do not want a volume name, simply press **Enter**.

   After the volume name is written to the partition, you are returned to the System Preparation menu.

Copy System Files

Use the Copy System Files function to

- copy MS-DOS system utilities to an MS-DOS partition.
- copy a preset system configuration file (CONFIG.SYS) to an MS-DOS partition.

When you choose to copy the MS-DOS system utilities to a partition, the following procedures are automatically performed:

- A subdirectory called \DOS is created in the root directory of the partition, and the MS-DOS system utilities are copied to the \DOS subdirectory.
- An AUTOEXEC.BAT file containing the command

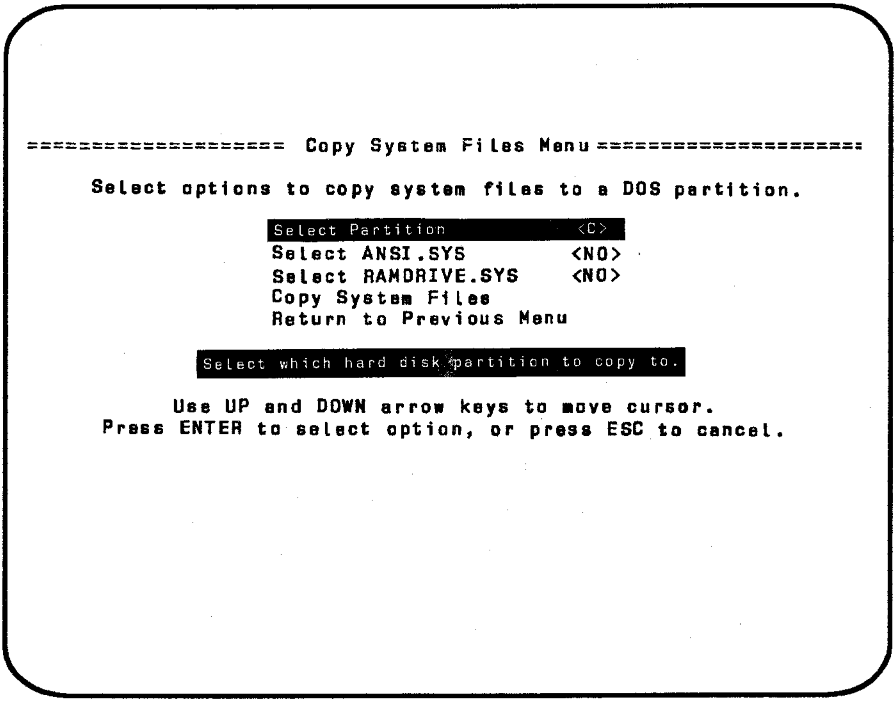
  ```
  PATH = \DOS
  ```

  is copied to the root directory of the partition.

When you choose to copy the preset CONFIG.SYS file to a partition, you can optionally add the device driver for a virtual disk (RAMDRIVE.SYS) to the system configuration file (CONFIG.SYS). A virtual disk is a portion of the PowerMate memory used as if it were a disk drive. For more information about the virtual disk device driver, refer to Appendix D.

To copy system files to a hard disk partition, follow these steps:

1. Select Copy System Files from the System Preparation menu. A screen similar to the following is displayed.

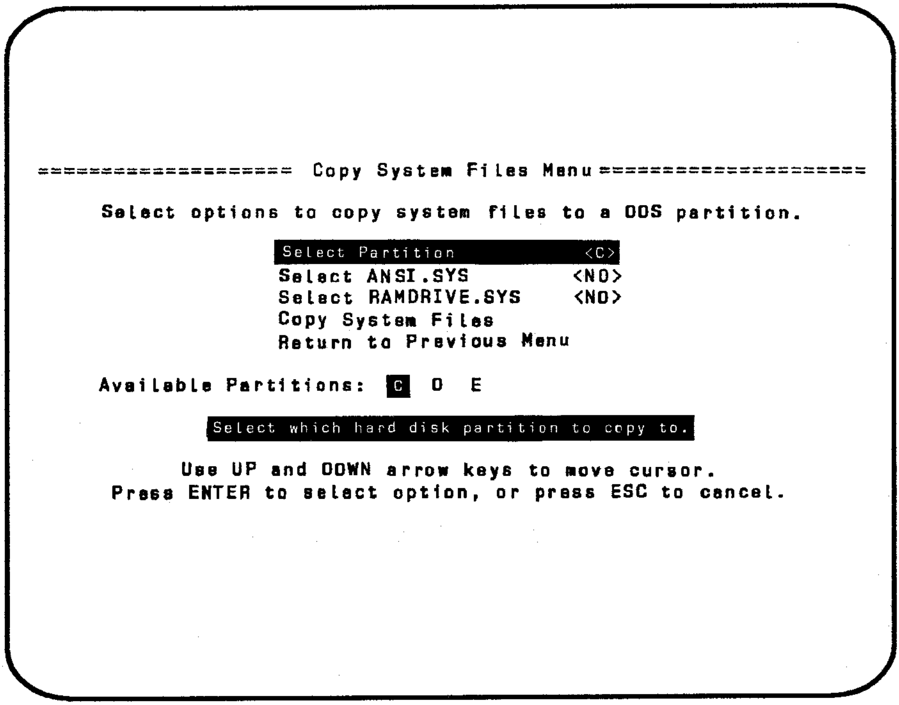


```
=================== Copy System Files Menu ====================

     Select options to copy system files to a DOS partition.

                  Select Partition          <C>
                  Select ANSI.SYS           <NO>
                  Select RAMDRIVE.SYS       <NO>
                  Copy System Files
                  Return to Previous Menu

           Select which hard disk partition to copy to.

            Use UP and DOWN arrow keys to move cursor.
      Press ENTER to select option, or press ESC to cancel.
```

2. Initially, the cursor highlights the topmost menu item. Use the up and down arrow keys to move the cursor among the various menu items.

   To change the value of a menu item, press **Enter** when it is highlighted.

3. If you chose the Select Partition option, the cursor moves to the lower part of the following screen.



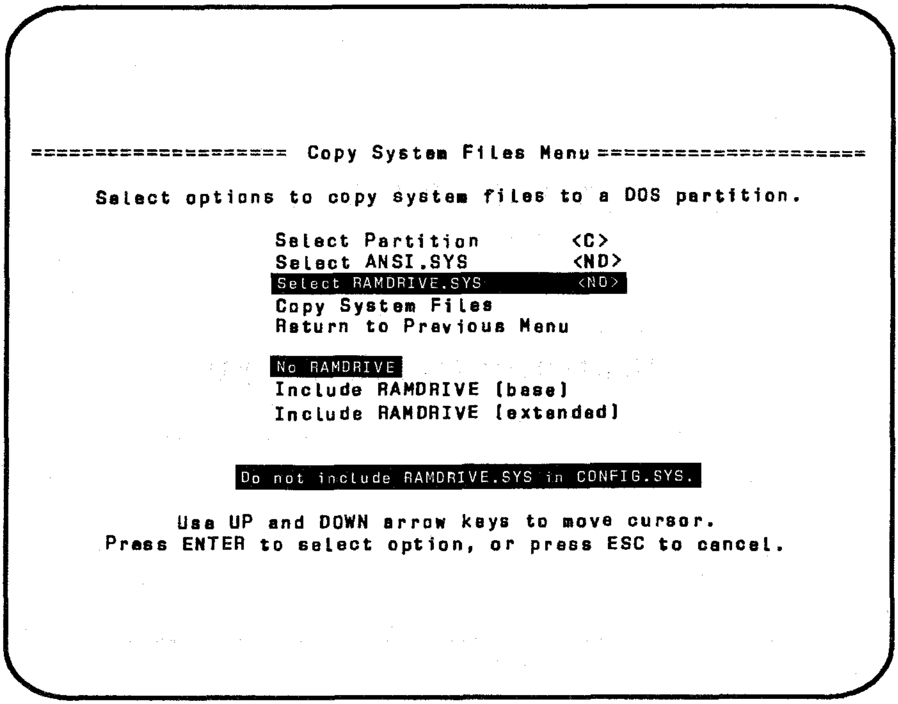
```
==================== Copy System Files Menu ====================

    Select options to copy system files to a DOS partition.

                  Select Partition          <C>
                  Select ANSI.SYS           <NO>
                  Select RAMDRIVE.SYS       <NO>
                  Copy System Files
                  Return to Previous Menu

    Available Partitions:  C  D  E

              Select which hard disk partition to copy to.

            Use UP and DOWN arrow keys to move cursor.
      Press ENTER to select option, or press ESC to cancel.
```

The drive letters displayed on your screen correspond to the available partitions on your hard disk(s).

To select a partition, use the cursor arrow keys to place the cursor on the desired partition. Press **Enter**. The cursor returns to the Select Partition option with the selected partition displayed.

4. If you specify YES for the Select ANSI.SYS option, ANSI.SYS will be included as a device specified in CONFIG.SYS. (Many applications programs require that ANSI.SYS be included in CONFIG.SYS.)

5. The Select RAMDRIVE.SYS option is used to determine whether or not to include RAMDRIVE.SYS as a device specified in CONFIG.SYS. If you select this option, the cursor moves to the lower part of the following screen.



```
==================== Copy System Files Menu ====================

     Select options to copy system files to a DOS partition.

                  Select Partition        <C>
                  Select ANSI.SYS         <NO>
                  Select RAMDRIVE.SYS     <NO>
                  Copy System Files
                  Return to Previous Menu

                  No RAMDRIVE
                  Include RAMDRIVE [base]
                  Include RAMDRIVE [extended]


               Do not include RAMDRIVE.SYS in CONFIG.SYS.

           Use UP and DOWN arrow keys to move cursor.
      Press ENTER to select option, or press ESC to cancel.
```

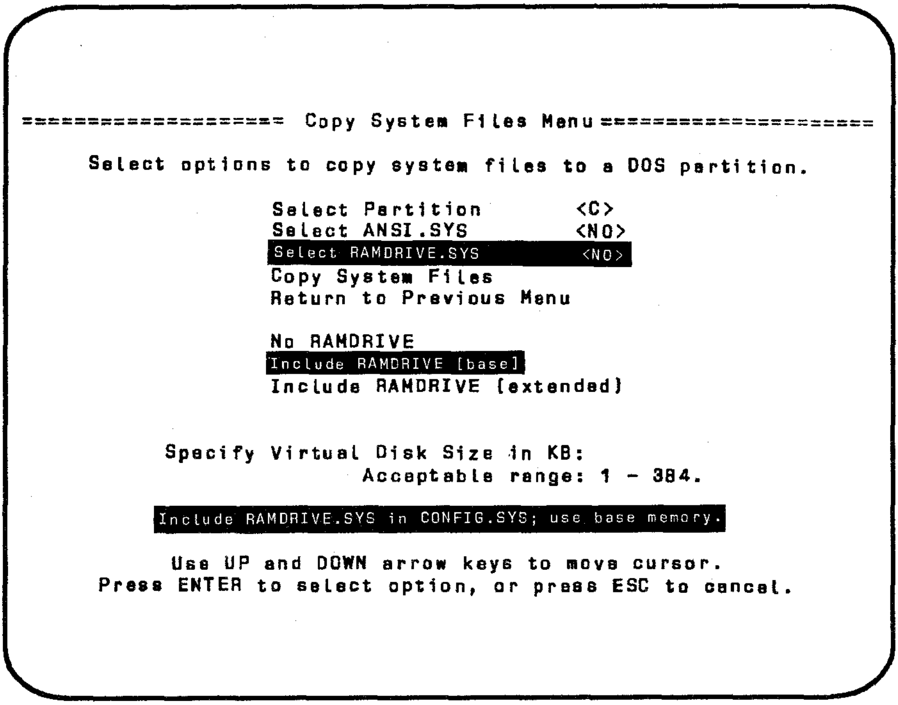
If you are creating a bootable partition and you want a virtual disk to be installed when the PowerMate is booted, select an option that includes RAMDRIVE.SYS (virtual disk device driver).

The virtual disk can be installed in nonextended memory or extended memory (memory above 1 MB). The extended memory option will only be displayed if your PowerMate has extended memory.

NOTE

The advantage of using extended memory for the virtual disk is that all the base memory will be free for use by applications that must reside in base memory.

If you select an option to include RAMDRIVE.SYS, you will be asked to specify the size of the virtual disk. A screen similar to the one following is displayed.



```
==================== Copy System Files Menu ====================

     Select options to copy system files to a DOS partition.

                 Select Partition          <C>
                 Select ANSI.SYS           <NO>
                 Select RAMDRIVE.SYS       <NO>
                 Copy System Files
                 Return to Previous Menu

                 No RAMDRIVE
                 Include RAMDRIVE [base]
                 Include RAMDRIVE [extended]


         Specify Virtual Disk Size in KB:
                    Acceptable range: 1 - 384.

         Include RAMDRIVE.SYS in CONFIG.SYS; use base memory.

          Use UP and DOWN arrow keys to move cursor.
      Press ENTER to select option, or press ESC to cancel.
```

The size of the virtual disk is specified in kilobytes (KB). Type a number that is within the range displayed and press Enter. Note: when specifying the size of a RAMDRIVE.SYS in extended memory, you can select a RAMDRIVE.SYS size that is greater that the amount of extended memory installed in your system. This is an allowable designation; in this case the virtual disk will occupy all of the extended memory.

6. When the menu items are correctly set, select Copy System Files to begin the copy process.

   When the copy process is completed, select Exit Copy System Files to return to the System Preparation menu.

   Once you have successfully completed the installation procedures, your PowerMate is ready to use. Refer to Section 2 if you need an introduction to using MS-DOS.

Appendix B

Keyboards

This appendix describes the use of the Alt key and special key combinations. To learn about more commonly used keys, see Section 2, "Getting Acquainted with MS-DOS."

This appendix also shows the five non-U.S. keyboard layouts. To use one of these layouts, issue the appropriate KEYBxx command as described in Section 4, "MS-DOS Commands."

THE Alt KEY

The Alt key temporarily shifts many keys to their Alt states. You can also use the Alt key simultaneously with the Ctrl and Del keys to cause the System Reset function described below.

The Alt key also allows you to enter any ASCII character code (as a numeric value) from 0 to 255 into the system from the numeric keypad. Simply hold down the Alt key and type the decimal value of the character desired, then release the Alt key.

If a value greater than 255 is typed, a modulo 256 result is created (i.e., the effective value equals the remainder yielded when the value entered is divided by 256).

SPECIAL KEY COMBINATIONS

At times, keys can be used in combinations to perform special functions. Press the keys simultaneously to perform these functions.

Line Feed

Ctrl-J or **Ctrl-Enter** produces a line feed.

System Reset

Ctrl-Alt-Del pressed in the order indicated (and held down) reloads the MS-DOS system. Use this to exit from infinite loops that may occur during an application program's execution (consult application program's manual first), or from situations where the system has been corrupted. This function is also referred to as a *warm boot*; it resets the computer without turning off the computer.

Key Click Volume

Ctrl-Alt-Plus (the + key on the numeric keypad) cycles the key click volume from no key click through seven levels of volume, then back to no key click. The volume defaults to "off" when the system boots.

CPU Speed

Ctrl-Alt-Minus (the - sign on the numeric keypad) toggles the CPU speed between high and low. The clock speed indicator on the front panel of the system unit light up when the higher speed is selected. Note that this key combination has no effect on systems having only one CPU speed.

Break

Ctrl-Pause/Break halts the execution of the current MS-DOS process, except when it immediately follows a PAUSE operation. Specifically, it invokes the keyboard routine that signals interrupt 1BH. The extended characters (AL = hex 00, AH = hex 00) will be returned for this keyboard operation.

Echo to Printer

Ctrl-Print Screen (or **Ctrl-P**) causes the system printer to print (echo) all characters that are subsequently output to the display. Press **Ctrl-P** or **Ctrl-Print Screen** again to cancel echo.

System Request

Alt-Print Screen/Sys Rq is a special-purpose key combination used by certain programs to cause a specific action to occur. The function of this key combination is dependent on the program that senses the System Request. In most cases, you will not use this key combination unless the program you are using has a defined function for it.

NON–U.S. KEYBOARD LAYOUTS

In addition to the standard U.S. keyboard layout, the following non–U.S. keyboard layouts are available:

- Belgian
- Canadian
- Danish
- Dutch
- French
- German
- Italian
- Latin American
- Norwegian
- Portuguese
- Spanish
- Swedish/Finish
- Swiss (Fr./Gr.)
- U. K. English

To select one of these keyboard layouts, use the KEYB command, described in Section 4, "MS–DOS Commands."

Some keys can produce up to four different characters, as shown on the keyboard layouts in this section. Use the key combinations in the following table to type the upper right and lower right characters. Use the shift key for the upper left characters. No additional keys are required for the lower left characters.

Country	**Lower right character except enhanced keyboard**	**Lower right character enhanced keyboard**	**Upper right character all keyboards**
Canada (Fr.)	Alt–Shift keys	Alt–Gr.	None
Denmark	Alt	Alt–Gr.	Alt–Shift keys
Finland	Alt	Alt–Gr.	Alt–Shift keys
Norway	Alt	Alt–Gr.	Alt–Shift keys
Sweden	Alt–Ctrl keys	Alt–Gr.	Alt–Shift keys
All others	Alt	Alt–Gr.	None

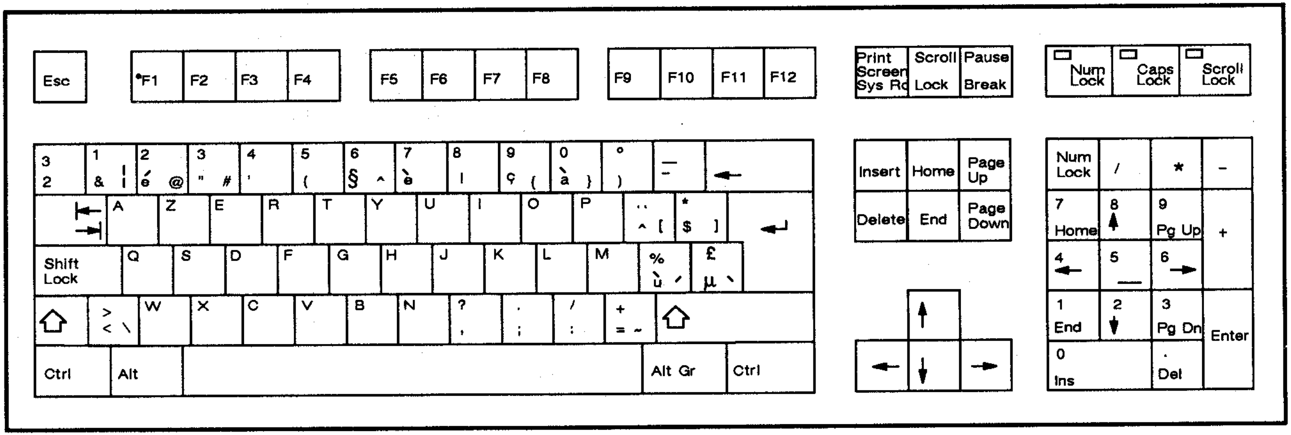


Figure B–1 Belgian Keyboard (101–key)

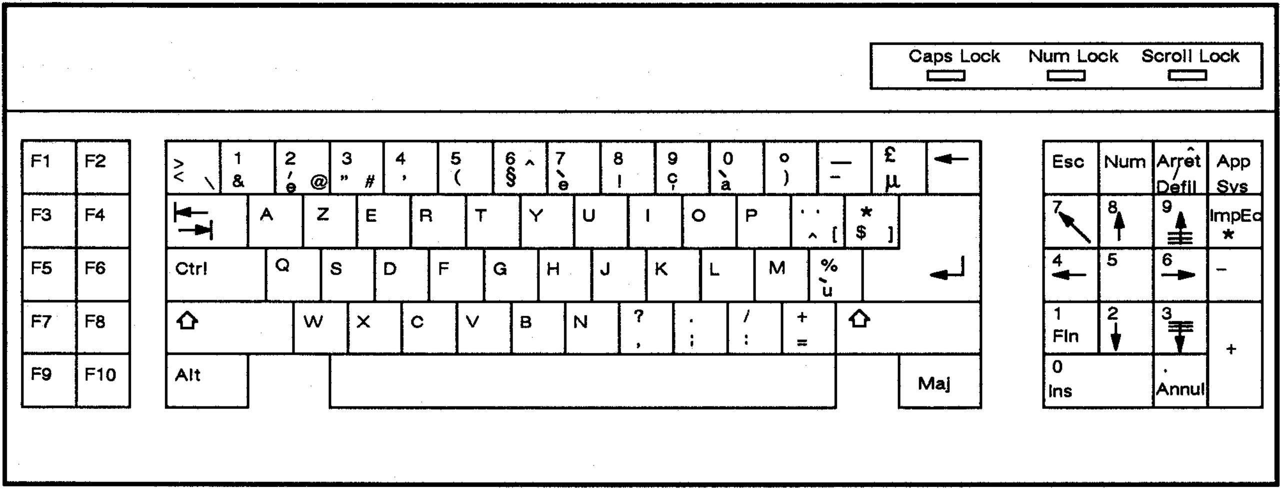


Figure B-2 Belgian Keyboard (84-key)

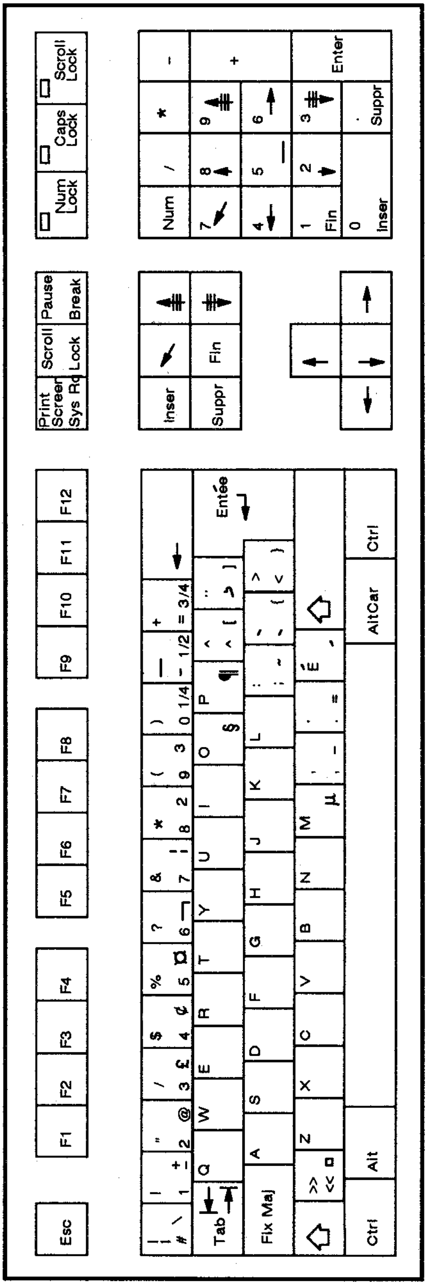


Figure B–3 Canadian Keyboard (101–key)

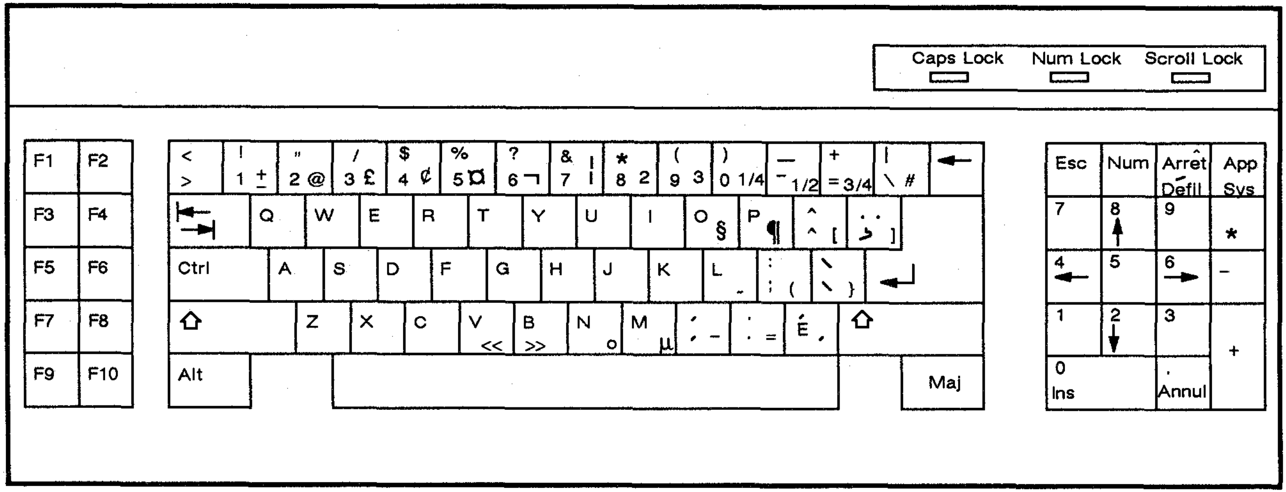


Figure B-4 Canadian Keyboard (84-key)

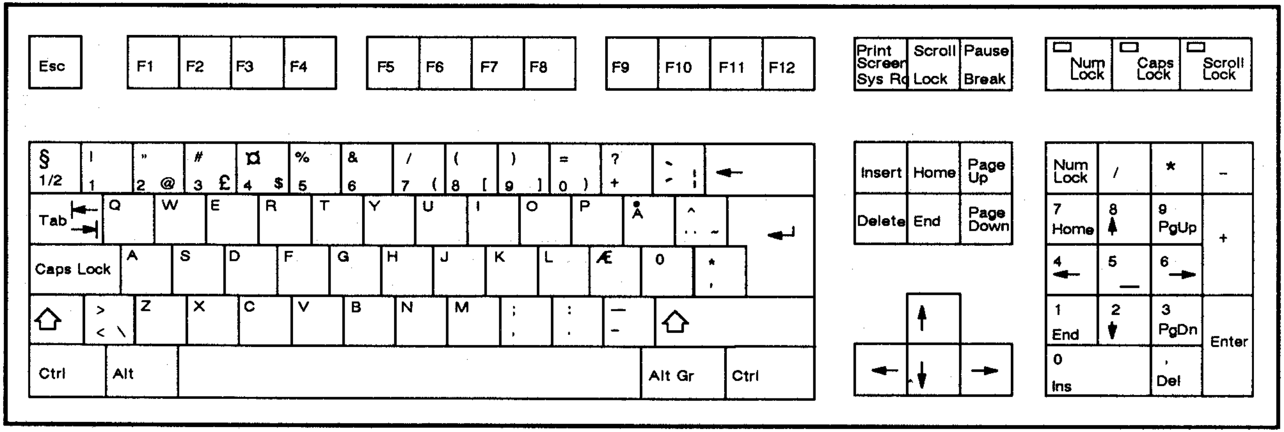


Figure B–5 Danish Keyboard (101–key)

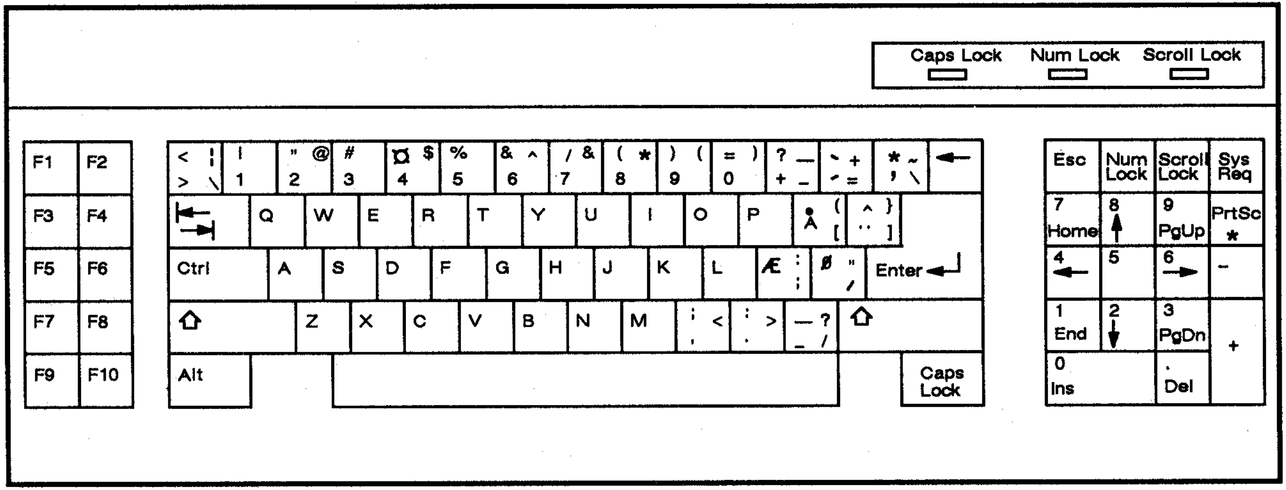


Figure B-6 Danish Keyboard (84-key)

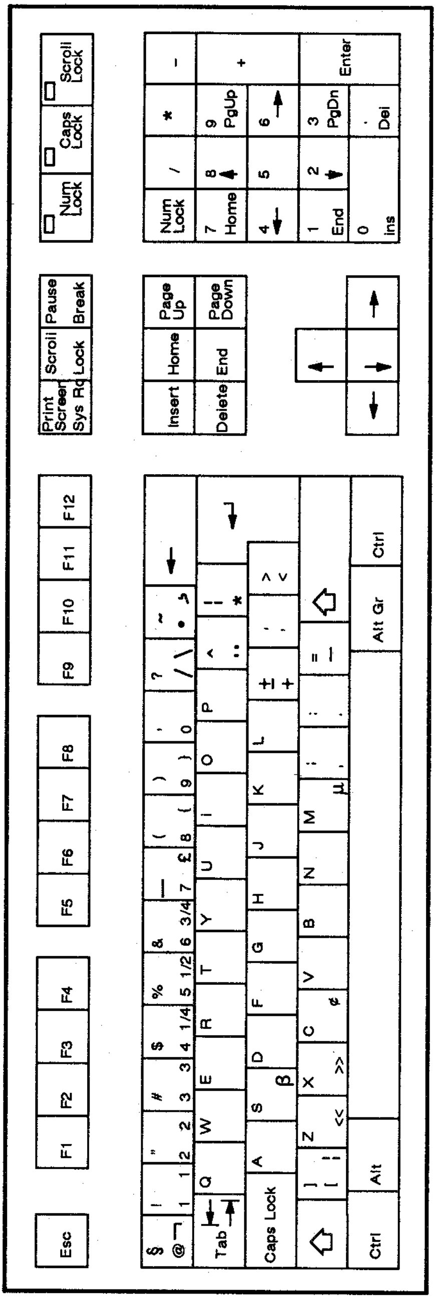


Figure B–7 Dutch Keyboard (101–key)

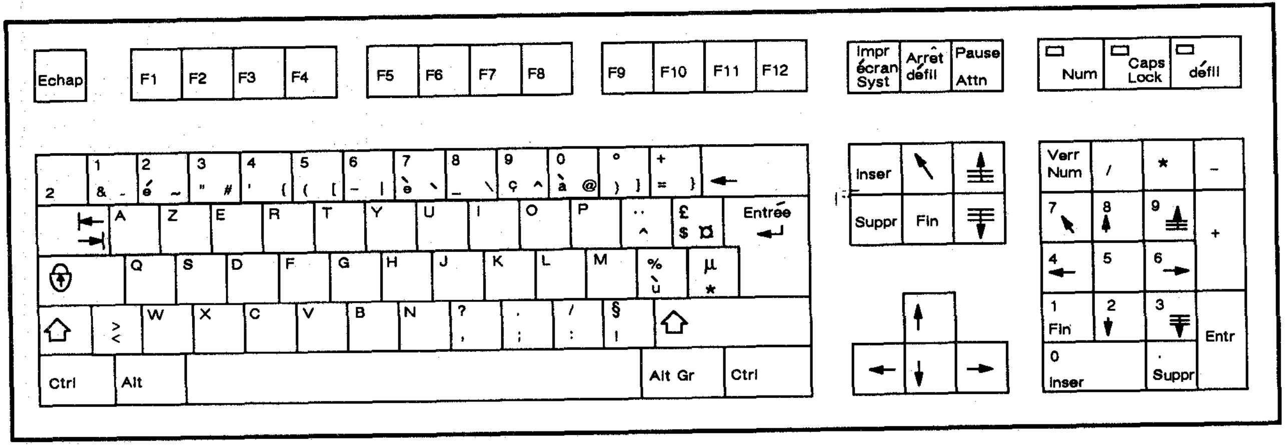


Figure B-8 French Keyboard (101-key)

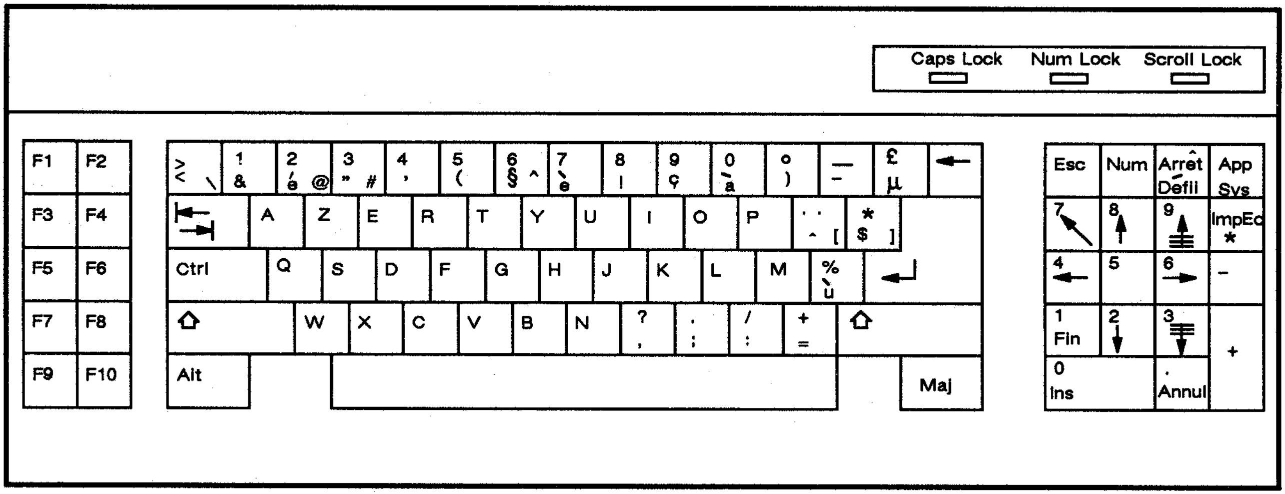


Figure B-9 French Keyboard (84-key)

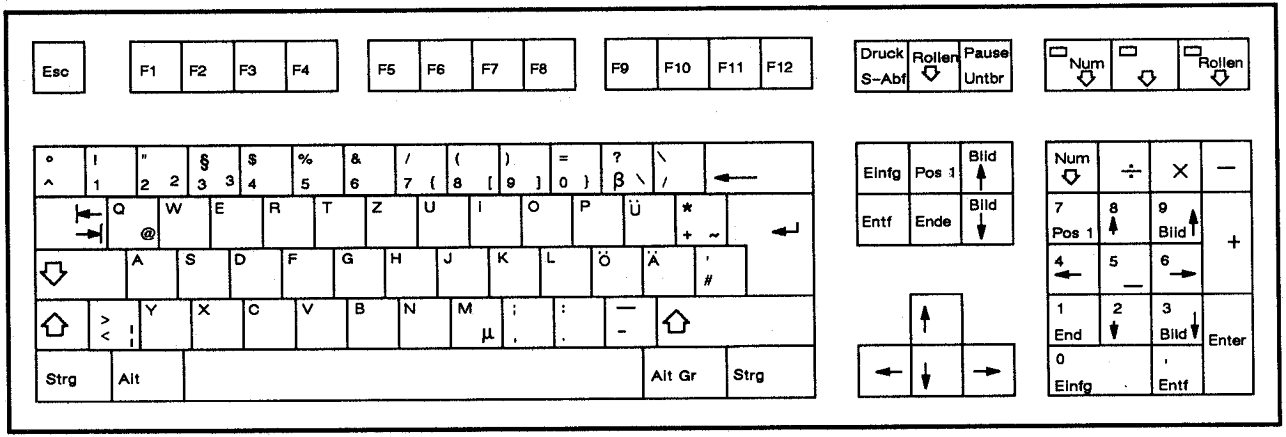


Figure B–10 German Keyboard (101-key)

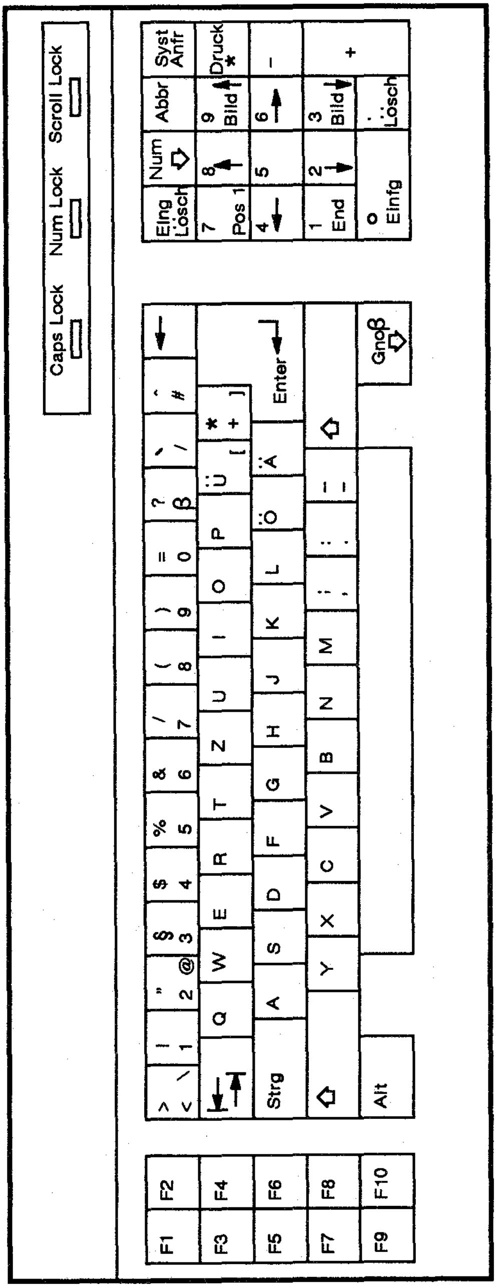


Figure B-11 German Keyboard (84-key)

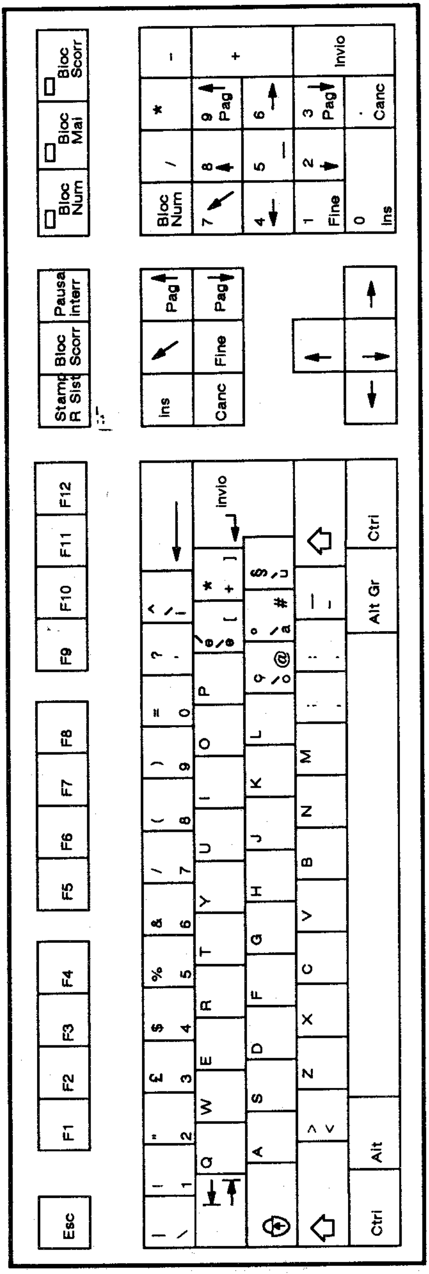


Figure B-12 Italian Keyboard (101-key)

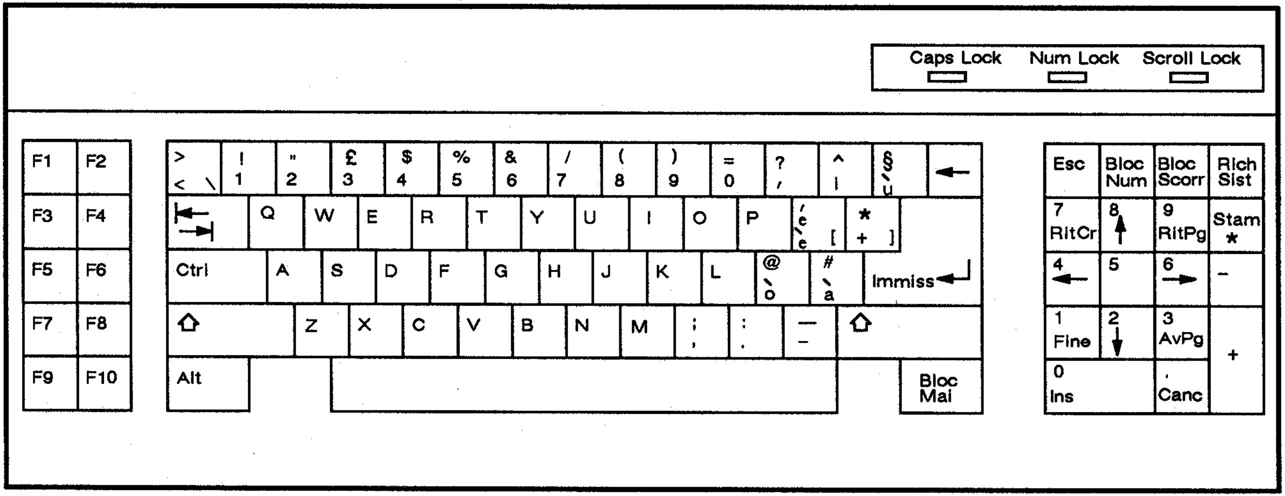


Figure B-13 Italian Keyboard (84-key)

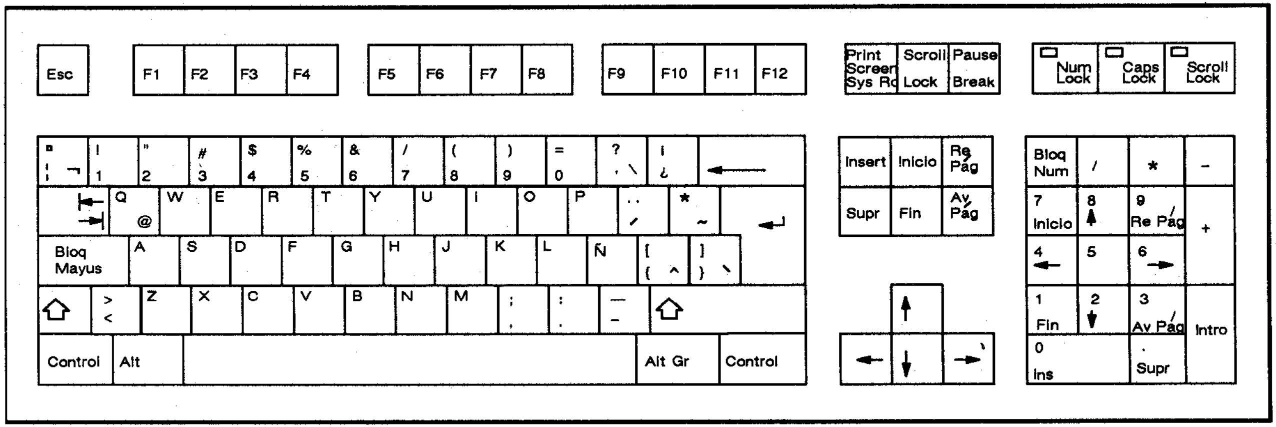


Figure B-14 Latin American Keyboard (101-key)

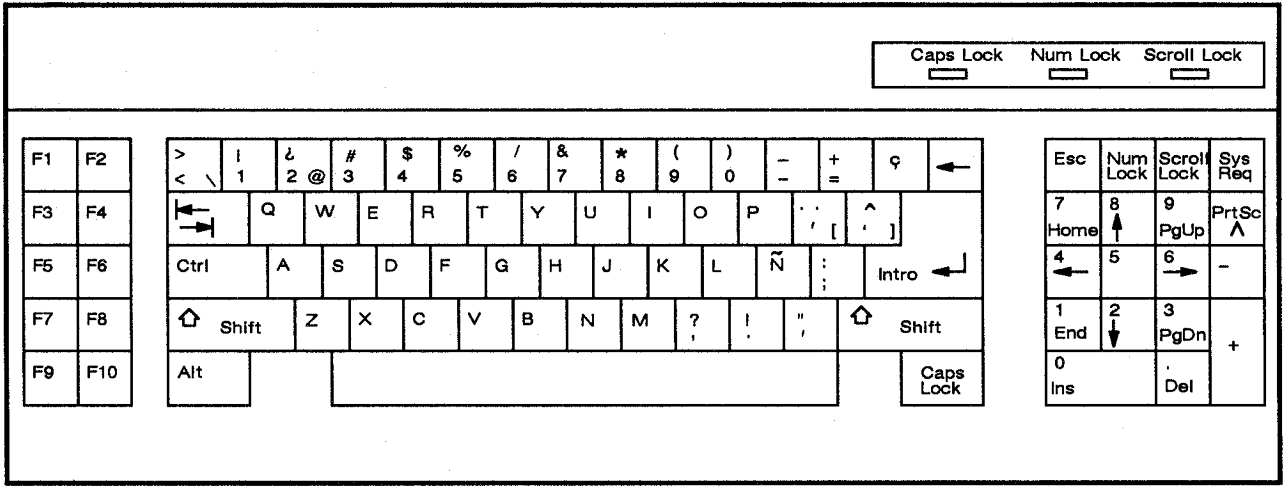


Figure B–15 Latin American Keyboard (84-key)

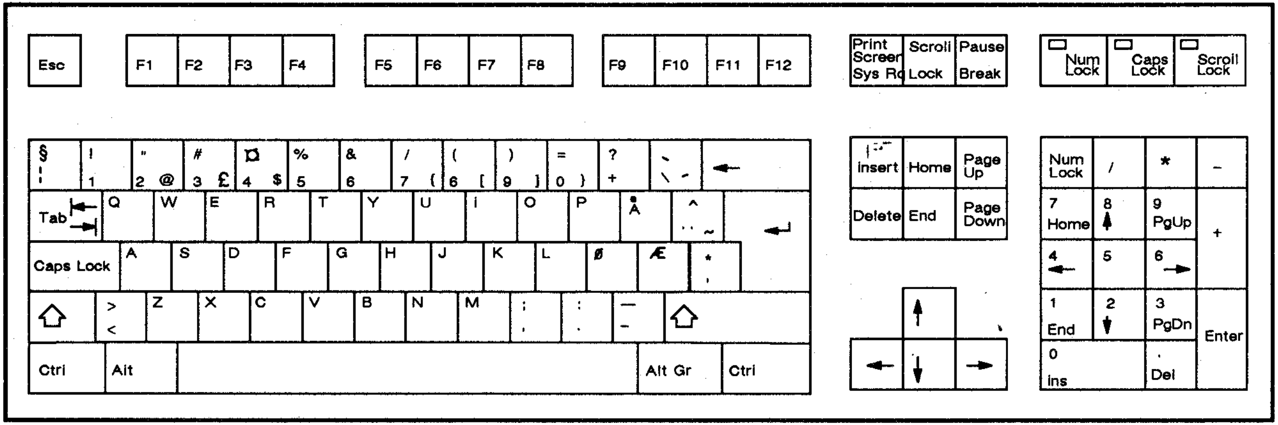


Figure B-16 Norwegian Keyboard (101-key)

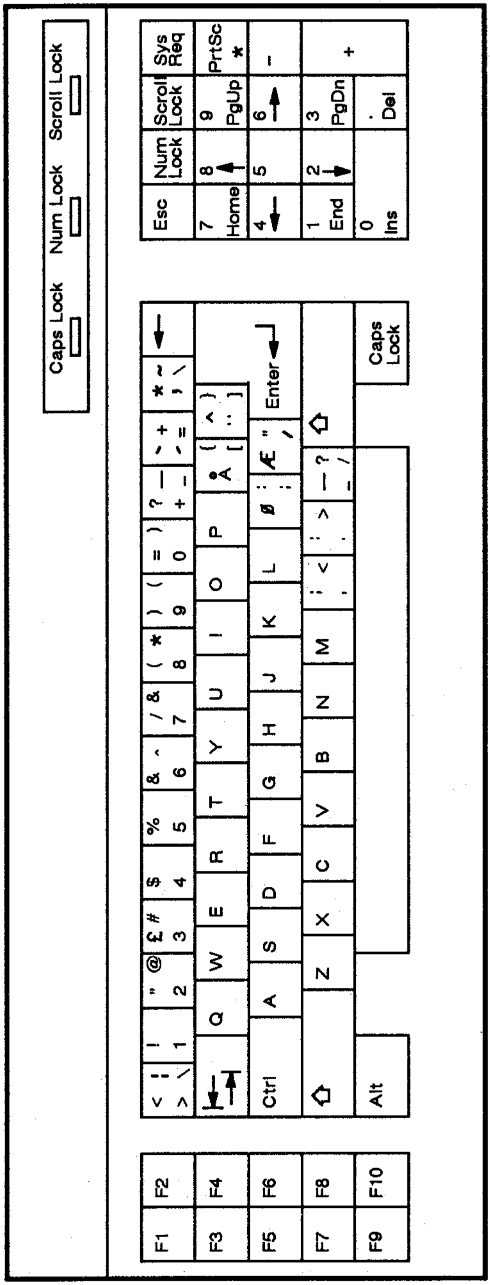


Figure B-17 Norwegian Keyboard (84-key)

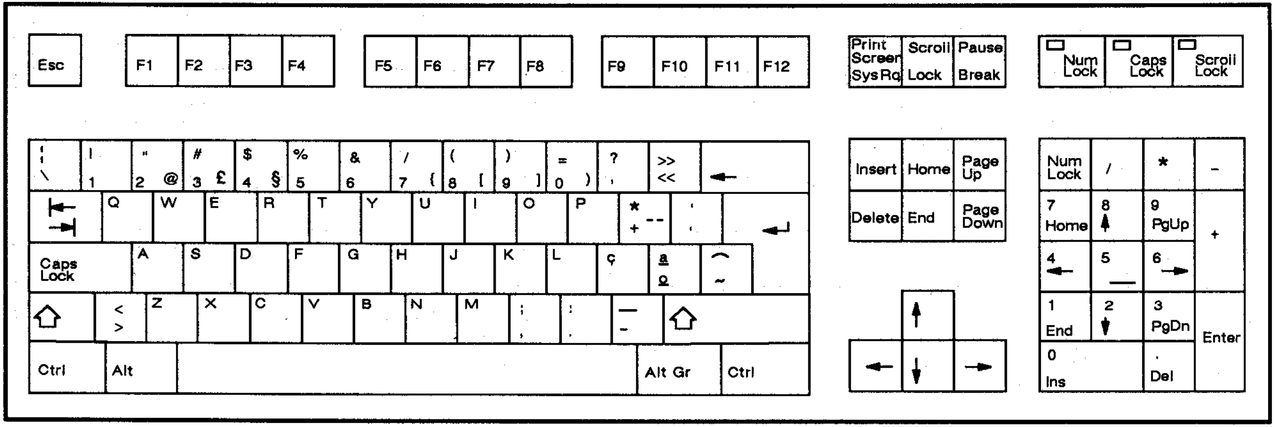


Figure B-18 Portuguese Keyboard (101-key)

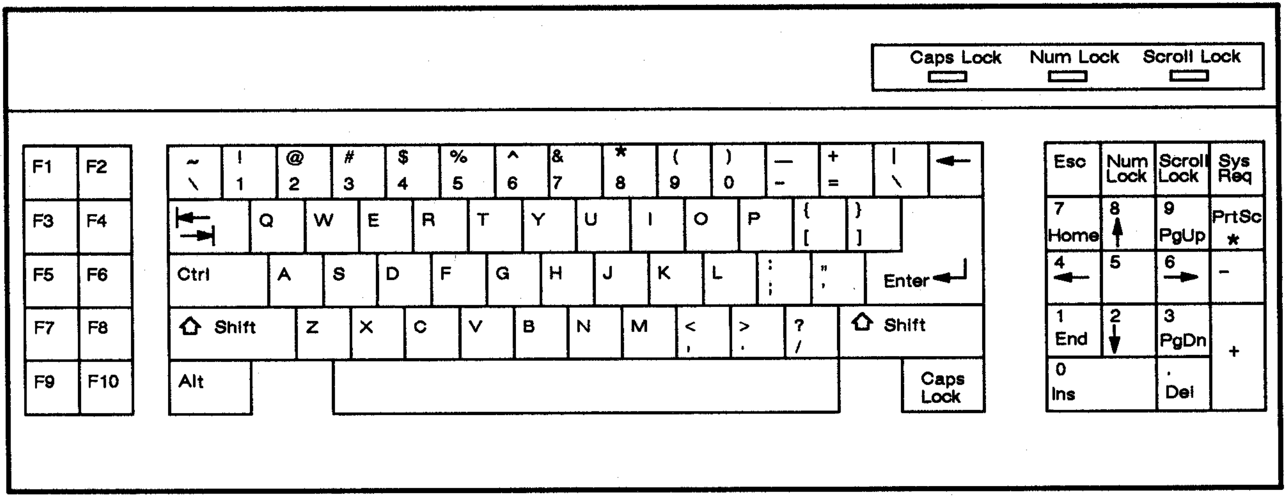


Figure B-19 Portuguese Keyboard (84-key)

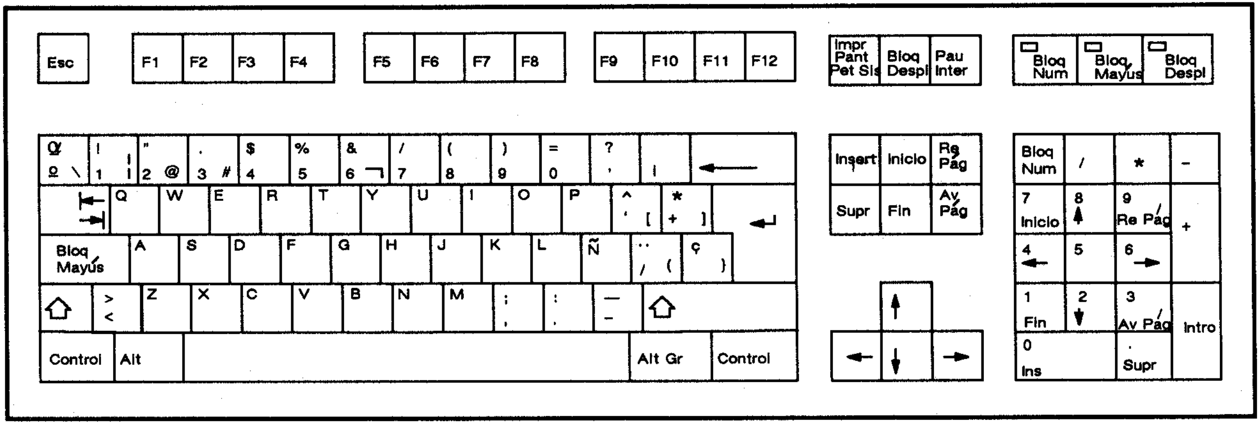


Figure B-20 Spanish Keyboard (101-key)

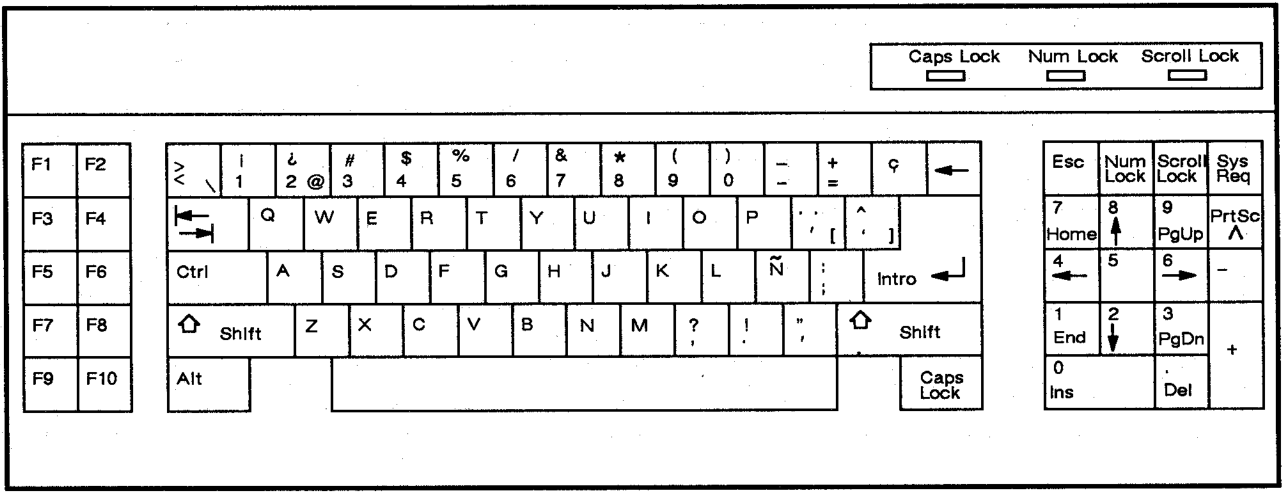


Figure B-21 Spanish Keyboard (84-key)

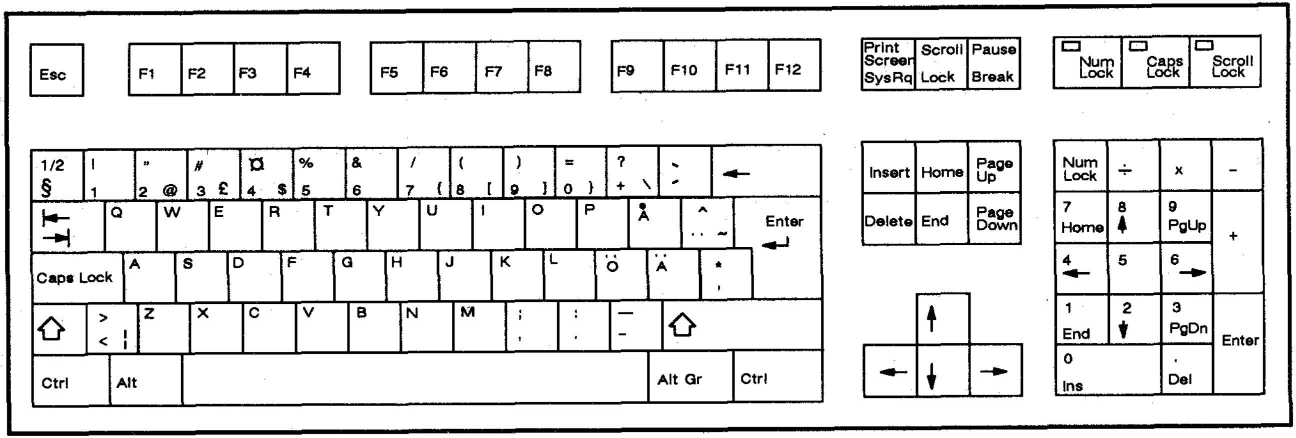


Figure B–22 Swedish Keyboard (101–key)

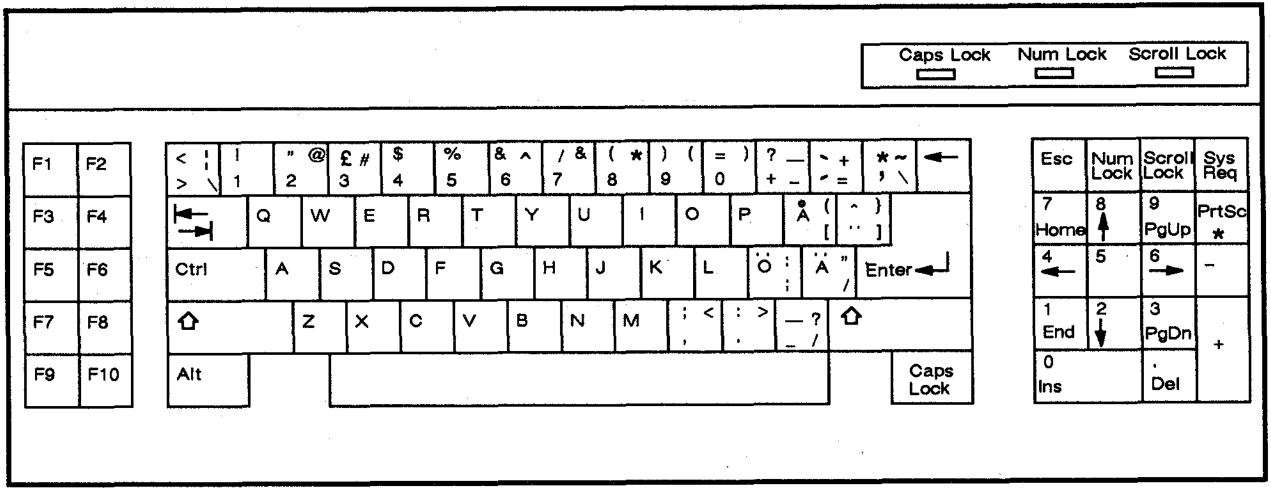


Figure B-23 Swedish Keyboard (84-key)

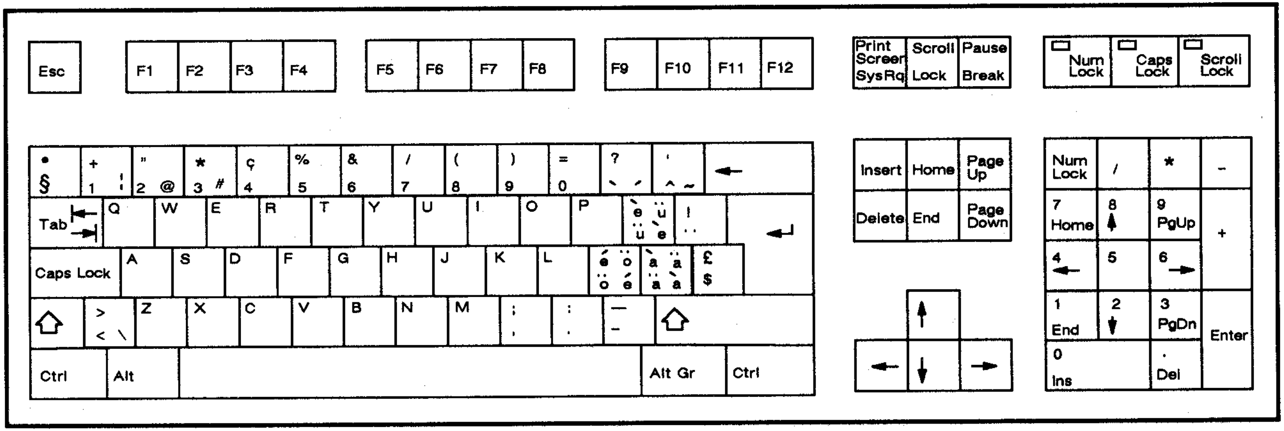


Figure B-24 Swiss (Fr./Gr.) Keyboard (101-key)

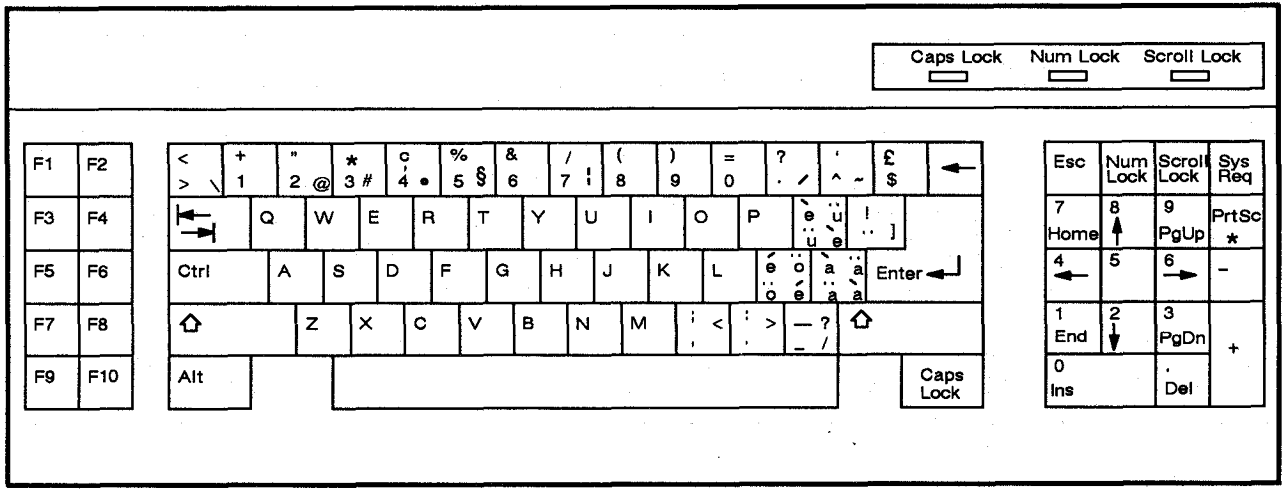


Figure B-25 Swiss (Fr./Gr.) Keyboard (84-key)

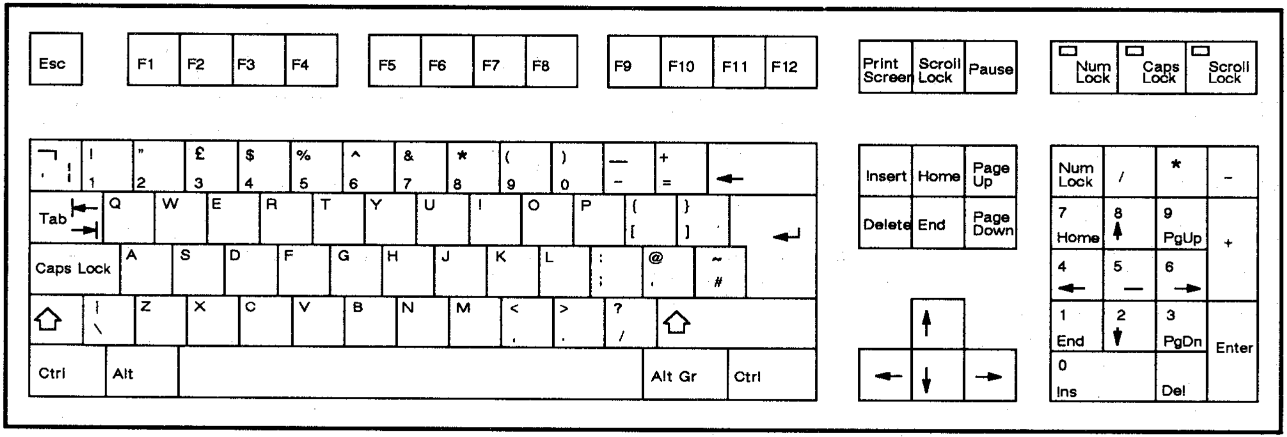


Figure B-26 U.K. English Keyboard (101-key)

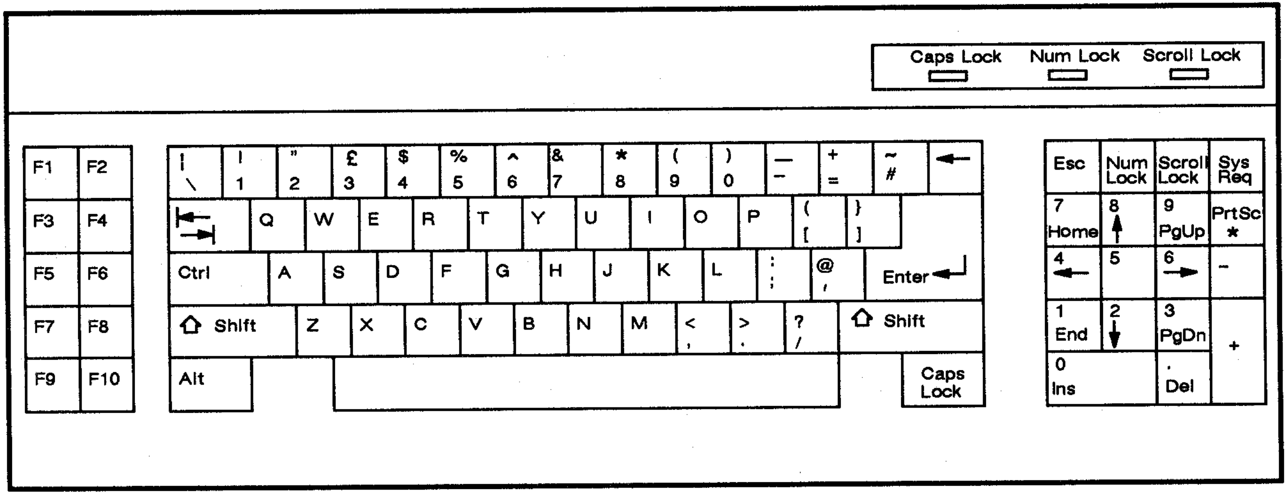


Figure B-27 U.K. English Keyboard (84-key)

Appendix C

How to Configure Your System

WHAT IS A CONFIGURATION FILE?

The configuration file CONFIG.SYS is a file that contains certain commands, which MS–DOS checks at startup. Each time you start MS–DOS, it searches the root directory of the drive in which it was started for a file named CONFIG.SYS.

The CONFIG.SYS file allows you to configure your system with a minimum of effort. For example, you can add device drivers to your system by including special commands in your CONFIG.SYS file.

If you used the System Setup procedure described in Appendix A, you already have a CONFIG.SYS file on your boot disk. If you have additional disks that need CONFIG.SYS files, or if you want to change the CONFIG.SYS file created by the System Setup utility, you can use the MS–DOS line editor, EDLIN.

CONFIG.SYS COMMANDS

The commands listed in Table C–1 can be used in the CONFIG.SYS file:

Table C-1 CONFIG.SYS Commands

COMMAND	WHAT IT DOES
BREAK	Sets Ctrl-C check.
BUFFERS	Sets the number of sector buffers.
COUNTRY	Allows for international time, date, and currency.
DEVICE	Installs the device driver in the system.
DRIVPARM	Defines parameters for block devices.
FCBS	Specifies the number of FCBs that can be open concurrently.
FILES	Sets the number of open files that can access certain MS-DOS system calls.
LASTDRIVE	Sets the maximum number of drives you may access.
SHELL	Begins execution of the shell from a specific file (usually COMMAND.COM).
STACKS	Supports the dynamic use of data stacks.

These commands are described in detail in the following pages.

BREAK

Purpose

Sets Ctrl–C check.

Syntax

BREAK=[ON] or

BREAK=[OFF]

Comments

Depending on the program you are running, you may use Ctrl–C to stop an activity (for example, to stop sorting a file). Normally, MS–DOS checks to see whether you have pressed Ctrl–C only while it is reading from the keyboard or writing to the screen or printer. Therefore, setting break to ON extends Ctrl–C checking to other functions, such as disk reading and writing.

Example

To turn off Ctrl–C checking, put the following line in your CONFIG.SYS file:

```
    BREAK=OFF
```

BUFFERS

Purpose

Allows you to set the number of disk buffers that MS-DOS allocates in memory at the time you start the system.

Syntax

BUFFERS=*x*

Default

Memory size	Buffers
For a base system	2
Any disk over 360 bytes	3
128K to 255K bytes	5
256K to 511K bytes	10
512K bytes or more	15

Comments

The *x* option is the number of disk buffers, from 2 to 255. A disk buffer is a block of memory that MS-DOS uses to hold data when reading or writing. For applications such as word processors, a number between 10 and 20 provides the best performance. If you plan to create a lot of subdirectories, you may even want to increase the buffers value to between 20 and 30. Remember, though, that buffers take up 512 bytes of space, so the more buffers you have, the less memory you will have available for applications.

Note

Feel free to experiment with different buffer settings to see how different values affect the way your personal computer operates.

Example

To create 20 disk buffers, put the following line in your CONFIG.SYS file:

```
BUFFERS=20
```

COUNTRY

Purpose

Country allows MS-DOS to use international time, date, currency, and case conversions.

Syntax

COUNTRY=*xxx*[,[*yyy*][,[*drive*:]*filename*]]

Options

xxx is the country code.

yyy is the code page for the country.

filename is a file containing country information.

Comments

This configuration command identifies to MS-DOS which country's character set you intend to use. Unless otherwise specified, United States settings are assumed.

If you do not specify filename, MS-DOS uses the COUNTRY.SYS file for country-specific information.

For a list of valid country codes, see Appendix E, How to Use Code Pages.

Example

The following example sets country to France (=033) and converts international currency, time, date, and case to French conventions:

```
COUNTRY=033
```

DEVICE=

Purpose

Installs the specified device driver on the system list.

Syntax

DEVICE=[*drive:*] [*path*]*filename*[*argument*]

Comments

The option *argument* includes any switches accepted by *filename*.

The standard installable device drivers provided with MS-DOS are ANSI.SYS, DISPLAY.SYS, DRIVER.SYS, PRINTER.SYS, and RAMDRIVE.SYS. For more information on these installable device drivers, see Appendix D, "Installable Device Drivers."

If you purchase a new device, like a mouse or a scanner, you generally will receive device driver software with that device. These installable device drivers can be installed using the DEVICE command. Once you have installed a device driver, make sure that the device driver is in the directory that you specify in any DEVICE commands.

Note

The device drivers COUNTRY.SYS and KEYBOARD.SYS are loaded automatically by MS-DOS. Do not try to load either of these with the DEVICE command. If you do, it will hang your system (that is, MS-DOS will not start).

Example

If you plan to use the ANSI escape sequences described in Appendix D, Installable Device Drivers, you should create a CONFIG.SYS file that contains the following command:

```
DEVICE=ANSI.SYS
```

This command causes MS-DOS to replace all keyboard input and screen output support with the ANSI escape sequences.

DRIVPARM

Purpose

This command allows you to define parameters for block devices when you start MS-DOS, overriding the original MS-DOS device driver settings.

Syntax

DRIVPARM=/D:number [/C][/F:factor]
[/H:heads] [/N][/S:sectors] [/T:tracks]

Comments

Setting DRIVPARM overrides any previous block device driver definitions. The following list describes how each switch is used:

SWITCH	FUNCTION
/D:number	Physical drive number, ranging from 0 to 255. This means that drive number 0=A, 1=B, 2=C, etc.
/C	Shows that change-line (door-lock) support is required. This means that the device driver will be able to tell whether the door of a floppy disk drive is open or closed. If the door is open, the device driver will assume that the drive does not have a disk in it yet.
/F:factor	Specifies the device type (form factor). The default value is 2. factor=form factor index, where 0 = 160/180K bytes, or 0 = 320/360K bytes 1 = 1.2 megabytes 2 = 720K bytes (3 1/2-inch disk) 3 = 8-inch single density 4 = 8-inch double density 5 = Hard disk 6 = Tape drive 7 = 1.44 megabytes (3 1/2-inch disk)

The default values for the following switches depend upon the form factor specified with the /f: switch. If you do not specify the /f: switch, DRIVPARM uses a default of 720K bytes (3 1/2-inch disk).

SWITCH	FUNCTION
/H:heads	Maximum head number, ranging from 1 to 99. The default value is 2.
/N	Specifies a nonremovable block device.
/S:sectors	Number of sectors per track, ranging from 1 to 99. The default value is 9.
/T:tracks	Number of tracks per side on the block device, ranging from 1 to 999.

Example

Suppose your computer has an internal tape drive on drive D that is configured at startup to write 20 tracks of 40 sectors per track. If you want to reconfigure this tape drive to write 10 tracks of 99 sectors each, you can put the following line in your CONFIG.SYS file:

```
DRIVPARM=/D:3 /F:6 /H:1 /S:99 /T:10
```

This command line overrides the default device driver settings, and supports a tape drive as drive D (in this case the logical and physical drive numbers are identical). This tape drive has 1 head and supports a tape format of 10 tracks and 99 sectors per track. (This assumes that the device driver for the tape device supports this configuration of tracks and sectors.) So to create a tape that you can read on another computer, one which can read only this alternate format, you might want to use this method.

FCBS

Purpose

Allows you to determine the number of File Control Blocks (FCBs) that can be open concurrently.

Syntax

FCBS=*x*,*y*

Comments

The *x* option represents the number of files that File Control Blocks can open at any one time. The default value for *x* is 4, but allowed values range from 1 to 255.

If an application tries to open more than *x* files by FCBs, the y option specifies the number of files opened by FCBs that MS-DOS cannot close automatically. The first *y* files opened by FCBs are protected from being closed. The default value is 0, but allowed values range from 1 to 255.

Note

The preferred method of accessing files is to use file handles instead of file control blocks. However, some older applications may require you to use the FCBS command in your CONFIG.SYS file. You should only use the command if an application requires you to do so.

Example

To open four files by FCBs and to protect the first two files from being closed, put the following line in your CONFIG.SYS file:

```
    FCBS=4,2
```

FILES

Purpose

Sets the number of open files that the MS–DOS system calls can access.

Syntax

FILES=*x*

Comments

The *x* option represents the number of open files that the system calls can access. The default value for *x* is 8, although allowed values range from 8 to 255. The maximum number of files that one program can have open at a time is 65,534.

MS–DOS system calls 2FH through 60H are compatible with the XENIX® operating system.

Note

A process must issue MS–DOS system call 67H to activate the extended handle.

Example

To let MS–DOS open 20 files at one time, put the following line in your CONFIG.SYS file:

```
    FILES=20
```

LASTDRIVE

Purpose

Sets the maximum number of drives you may access.

Syntax

LASTDRIVE=*x*

Comments

The *x* value can be any letter from A to Z (1 to 26 drives). The default is E (5 drives).

The *x* value represents the last valid drive that MS–DOS will accept. The minimum number is equal to the number of drives you have installed on your computer.

This command is useful only in a network environment. At start–up, MS–DOS recognizes five drives you have on your system. To make any extra drives defined by LASTDRIVE valid, a network redirection must occur.

Note that MS–DOS allocates a data structure for each drive that you specify, so you should not specify more drives than are necessary.

Example

The following command sets the last drive to M, unless you have added an external logical device with DRIVER.SYS. For information about DRIVER.SYS, see Appendix D, Installable Device Drivers.

```
    LASTDRIVE=M
```

SHELL

Purpose

Begins execution of the SHELL (top–level command processor) from a fil(defined by the specified pathname.

Syntax

SHELL=[*drive*:] [*path*]*filename*

Comments

The default command processor for MS–DOS is COMMAND.COM.

The filename option specifies the program that MS–DOS uses as a command processor. Instead of reading the standard COMMAND.COM, MS–DOS starts the processor specified in filename.

System programmers who write their own command processors (instead of using the MS–DOS file, COMMAND.COM) should use the SHELL command to specify the name of their SHELL program.

MS–DOS sets the COMSPEC environment variable equal to the drive: path and filename specified on the SHELL command line. This setting overrides the default value for COMSPEC (the drive and pathname of the command processor initially used to start MS–DOS). The operating system uses the COMSPEC environment setting to determine which file to use when reloading any transient part of the command processor.

Note

The SHELL command does not accept switches. However, if the new command processor does accept switches, you can include those switches in this syntax.

For example, suppose SHELL=NEWCMDP.COM and suppose NEWCMDP.COM accepts the /C, /P, /E switches. You can include any of these switches in the SHELL command line. Thus, the following would be a valid command:

```
SHELL=NEWCMDP.COM /P
```

Example

The following command uses the file \BIN\NEWSHELL as the command processor:

```
SHELL=\BIN\NEWSHELL
```

STACKS

Purpose

Supports the dynamic use of data stacks for hardware interrupt handling.

Syntax

STACKS=*n*,*s*

Default

STACKS

9,128

Comments

The *n* value is the number of stacks. The valid values for *n* range from 0 to 64.

The *s* value is the size of each stack. The valid values for *s* range from 0 to 512. When there is a hardware interrupt, MS–DOS allocates one stack from *n* stack specified. When STACKS=0,0, MS–DOS will not switch stacks at interrupt time.

Examples

If you want to allocate eight stacks of 512 bytes each, you would include the following command in your CONFIG.SYS file:

```
    STACKS=8,512
```

Appendix D

Installable Device Drivers

INTRODUCTION

Device drivers are programs that let the operating system recognize devices that are not part of the computer. Examples of devices are a modem, a printer, a mouse and an external disk drive. Some device drivers are already installed with MS–DOS. Other device drivers, called installable device drivers, come with MS–DOS for you to install if you need them.

This appendix describes the installable device drivers provided with MS–DOS:

- ANSI.SYS loads the ANSI character set.
- DISPLAY.SYS supports code page switching on the console device.
- DRIVER.SYS supports external floppy disk drives.
- PRINTER.SYS provides code page support for PRN, LPT1, LPT2, and LPT3.
- RAMDRIVE.SYS supports one or more RAM (virtual) drives.

For more information about the device configuration command, which is used to install these device drivers, see Appendix C, "How to Configure Your System."

ANSI.SYS

ANSI.SYS is a character device driver that causes the computer to emulate American National Standards Institute (ANSI) standard terminals. An ANSI escape sequence is a series of characters (beginning with an escape character or keystroke) that you can use to define functions for MS–DOS. Specifically, you can change graphics functions and affect the movement of the cursor.

In the escape sequences that follow,

- *Pn* represents the numeric parameter, a decimal number that you specify with ASCII digits.
- *Ps* represents the selective parameter, a decimal number that you use to select a subfunction. You may specify more than one subfunction by separating the parameters with semicolons.
- *Pl* represents the line parameter, a decimal number that you specify with ASCII digits.
- *Pc* represents the column parameter, a decimal number that you specify with ASCII digits.

MS–DOS uses a default value when you do not specify a value or when you specify zero.

Cursor Functions

The following escape sequences affect the position of the cursor on the screen.

CUP--Cursor Position

ESC [*Pl* ; *Pc* H

HVP--Horizontal & Vertical Position

ESC [*Pl* ; *Pc* f

CUP and HVP move the cursor to the position specified by the parameters. The first parameter specifies the line number; the second, the column number. The default value for *Pl* and *Pc* is 1. When no parameters are specified, the cursor moves to the home position (the upper left-hand corner of the screen).

CUU--Cursor Up

ESC [*Pn* A

This sequence moves the cursor up one line without changing columns. The value of *Pn* sets the number of lines moved. The default value is 1. If the cursor is already on the top line, MS-DOS ignores the CUU sequence.

CUD--Cursor Down

ESC [*Pn* B

This sequence moves the cursor down one line without changing columns. The value of *Pn* sets the number of lines moved. The default value is 1. If the cursor is already on the bottom line, MS-DOS ignores the CUD sequence.

CUF--Cursor Forward

ESC [*Pn* C

The CUF sequence moves the cursor forward one column without changing lines. The value of *Pn* sets the number of columns moved. The default value is 1. If the cursor is already in the far right column, MS-DOS ignores the CUF sequence.

CUB--Cursor Backward

ESC [*Pn* D

This escape sequence moves the cursor back one column without changing lines. The value of *Pn* sets the number of columns moved. The default value is 1. If the cursor is already in the far left column, MS-DOS ignores the CUB sequence.

DSR--Device Status Report

ESC [6 n

The console driver outputs an RCP sequence when it receives the DSR escape sequence.

SCP--Save Cursor Position

ESC [s

The console driver saves the current cursor position. This position can be restored by the RCP 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.

Erase Functions

The following escape sequences affect erase functions.

ED--Erase Display

ESC [2 J

The ED sequence erases the screen. The cursor then goes to the home position.

EL--Erase Line

ESC [K

This sequence erases from the cursor to the end of the line (including the cursor position).

Modes of Operation

The following escape sequences affect screen graphics, but they work only if your monitor supports graphics functions.

SGR--Set Graphics Rendition

ESC [*Ps* ; ... ; *Ps* m

The SGR escape sequence calls the graphics functions specified by the parameters described in the following list. These functions remain until the next occurrence of an SGR escape sequence.

PARAMETER	FUNCTION
0	All attributes off
1	Bold on
2	Faint on
3	Italic on
5	Blink on
6	Rapid blink on
7	Reverse video on
8	Concealed on
30	Black foreground
31	Red foreground
32	Green foreground
33	Yellow foreground
34	Blue foreground
35	Magenta foreground
36	Cyan foreground
37	White foreground
40	Black background
41	Red background
42	Green background
43	Yellow background
44	Blue background
45	Magenta background
46	Cyan background
47	White background
48	Subscript
49	Superscript

Parameters 30 through 47 meet the ISO 6429 standard.

SM--Set Mode

ESC [= *Ps* h ESC [= h ESC [= 0 h ESC[? 7 h

The SM escape sequence changes the screen width or type to one of the following:

PARAMETER	FUNCTION
0	40 x 25 black and white
1	40 x 25 color
2	80 x 25 black and white
3	80 x 25 color
4	320 x 200 color
5	320 x 200 black and white
6	640 x 200 black and white
7	Wraps at the end of each line

RM--Reset Mode

ESC [= *Ps* l ESC [= l ESC [= 0 l ESC [? 7 l

Parameters for RM are the same as for SM (Set Mode), except parameter 7 resets the mode that causes wrapping at the end of each line.

DISPLAY.SYS

Purpose

DISPLAY.SYS is an installable device driver that supports code page switching for the console device.

Syntax

DEVICE=[*drive*:] [*path*]DISPLAY.SYS
con[:]=[*type*[,*hwcp*] [,*n*,*m*]]

Options

type — The display adapter in use. Valid values include MONO, CGA, options EGA,. and LCD.

hwcp — The code page supported by the hardware. The following values are allowed:

437 (United States)
850 (Multilingual)
860 (Portugal)
863 (French–Canadian)
865 (Norway)

n — The number of additional code pages that can be supported. This number is dependent on the hardware. MONO and CGA do not support other fonts, so *n* must be 0. EGA can be 2. LCD can be 1.

m — The number of sub–fonts that are supported for each code page.

DRIVER.SYS

Purpose

DRIVER.SYS is an installable device driver that supports external floppy disk drives.

Syntax

```
DEVICE=DRIVER.SYS /D:number [/C]
[/F:factor] [/H:heads] [/N] [/S:sectors]
[/T:tracks]
```

Switches

/D:number Physical drive number, ranging from 0 to 255. The first physical floppy disk drive is number 0, and is referenced from the MS-DOS command line as drive A. Drive number 1 is the second physical floppy disk drive. Drive 2 is the third, which must be external.

/C Shows that change-line (doorlock) support is required. This means that the device driver will be able to tell whether the door of a floppy disk drive is open or closed. If the door is open, the device driver will assume that the drive does not have a disk in it yet.

/F:factor Specifies the device type (form factor). The default value is 2.

factor=form factor index, where

0 = 160/180K bytes, or
0 = 320/360K bytes
1 = 1.2 megabytes
2 = 720K bytes (3 1/2-inch disk)
3 = 8-inch single density
4 = 8-inch double density
5 = Hard disk
6 = Tape drive
7 = 1.44 megabytes (3 1/2-inch disk)

/H:heads Maximum head number, ranging from 1 to 99. The default value is 2.

/N Specifies a nonremovable block device. A fixed disk is an example of a nonremovable block device.

/S:sectors Number of sectors per track, ranging from 1 to 99. The default value is 9.

/T:tracks Number of tracks per side on the block device, ranging from 1 to 999. The default value is 80.

Example

To add an external 720K–byte drive to your computer, youwould include the following line in the CONFIG.SYS file:

```
DEVICE=DRIVER.SYS /D:02
PRINTER.SYS
```

Purpose

PRINTER.SYS is an installable device driver that supports code page switching for parallel ports LPT1, LPT2, and LPT3. (The port name PRN may be substituted for LPT1 to refer to the first parallel port.)

Syntax

DEVICE=[*drive*:] [*path*]PRINTER.SYS LPT*x*=
[*type*[,*hwcp*[,...]] [,*n*]]

Options

type The printer in use.

hwcp The code page supported by the hardware. The following values are allowed:

437 (United States)
850 (Multilingual)
860 (Portugal)
863 (French–Canadian)
865 (Norway)

n The number of additional code pages that can be supported. This number is dependent on the hardware.

RAMDRIVE.SYS

Purpose

RAMDRIVE.SYS is an installable device driver that lets you use a portion of your computer's memory as if it were a hard disk. This memory area is called a RAM disk and is sometimes referred to as a virtual disk.

Syntax

DEVICE=RAMDRIVE.SYS [*disksize*] [*sector–size*] [*entries*] [/E]

or

DEVICE=RAMDRIVE.SYS [*disksize*] [*sector–size*] [*entries*] [/A]

Comments

RAM disks are much faster than hard disks because the information they contain is always loaded into memory. If your computer has extended memory installed (starting at the one megabyte boundary), or if you have an extended memory board that meets the Lotus/Intel/Microsoft Expanded Memory Specification, you can use this extended memory for one or more RAM disks. Otherwise, RAMDRIVE.SYS places RAM disks in low memory.

Note

The command DEVICE=RAMDRIVE.SYS increases the size of MS–DOS resident in memory.

Options

disksize Specifies the disk size in kilobytes. The default size is 64K bytes; the minimum size is 16.

sectors Specifies the sector size in bytes. The default size is 128 bytes. The following values are allowed: 128, 256, 512, and 1024 bytes.

entries Specifies the number of root directory entries. The default value is 64; the minimum, 4; the maximum, 1024.

RAMDRIVE.SYS adjusts the value of entries to the near est sector boundary. For example, if you specify a value of 25 when the sector size is 512 bytes, the 25 will be rounded up to 32, which is the next multiple of 16 (there are sixteen 32–byte directory entries in 512 bytes).

/E Lets you use extended memory (above one megabyte) as a RAM disk if it has been installed. If you use this switch, you cannot use the /A switch. It is recommended that you use the /E switch.

/A Lets you use an extended memory board that meets the Lotus/Intel/Microsoft Expanded Memory Specification for a RAM drive – if that board has been installed. If you use this switch, you cannot use the /E switch.

Note

When you reset or turn off the power on your computer, the information stored in RAM disks is lost.

VDISK.SYS

Purpose

VDISK.SYS is a device driver that acts like a disk by using a portion of the computer's memory for storing data. These disks are called virtual disks (VDISKS).

Syntax

DEVICE=VDISK.SYS [*bbb*] [*sss*] [*ddd*] [/E[:*m*]]

Comments

Install the VDISK.SYS driver by entering the DEVICE=VDISK.SYS command in your CONFIG.SYS file.

Table D-3 lists the parameters, value ranges, and default values for the DEVICE=VDISK.SYS command.

Table D-3 VDISK.SYS Command Parameters

PARAMETERS	ALLOWABLE VALUES	DEFAULT VALUE	COMMENTS
[bbb] - the virtual size in kilobytes	1 - (amount of available memory in your computer)	64	VDISK can adjust this value. See Note 1.
[sss] - the sector size in bytes	128, 256, 512	128	VDISK adjusts the sector size to 128 if a value other than 128, 256, or 512 is entered.
[ddd] - the number of directory entries that the virtual disk can hold.	2-512	64	VDISK can adjust this value. See Note 2.
[m] - maximum number of sectors of data VDISK transfers at a time.	1,2,3,4,5,6,7,8	8	Data transfer can result in lost interrupts. See Note 3.

Options

Use the /E switch to instruct VDISK to use extended memory. Extended memory is memory at or above 1 megabyte (MB).

The /E switch causes VDISK to place the virtual disk buffer in extended memory and install the device driver as a part of DOS.

The /E switch functions only if your computer has extended memory. If you use the /E switch with computers that do not have extended memory, VDISK issues an error message, and does not install the virtual disk.

Note 1: [*bbb*] Adjustment

VDISK adjusts the virtual disk size [bbb] under the following conditions:

- If there is fewer than 64 kilobytes (KB) of available memory, VDISK issues an error message and does not install the virtual disk.
- If you specify a virtual disk size that is less than 1 kilobyte or is greater than the amount of memory on your computer, VDISK uses the default value of 64 kilobytes.
- If you specify a virtual disk size that leaves less than 64 kilobytes of memory, VDISK adjusts the size of the virtual disk so that 64 kilobytes of available memory remains.

Note 2: [*ddd*] Adjustment

VDISK adjusts your directory entry [ddd] under the following conditions:

- If your entry is not a multiple of the sector size [sss], then VDISK adjusts the number of directory entries upwards to the nearest sector size boundary.

  For example, if you specify 10 directory entries (files) and your sector size is 128, VDISK generates 12 directory entries. The 12 entries at 32 bytes each round up to 384, or three 128–byte sectors.

- If the value is too small to hold the file allocation table, the directory, and two additional sectors, VDISK adjusts the size downward, one sector at a time.

  If the directory size reaches one sector and is still too large for the specified virtual disk size, VDISK issues an error message and does not install the virtual disk.

  VDISK uses one of the directory entries to hold a volume label.

Note 3: Interrupt losses

DOS suspends interrupt servicing during data transfers when VDISK operates in extended memory. If frequent interrupts occur, for example, during high speed communications, some interrupts can be lost.

To prevent the loss of interrupts, try installing VDISK in nonextended memory. If installing VDISK in nonextended memory solves the problem, adjust *m* (in extended memory) until no interrupts are lost.

If an *m* of 1 and *sss* of 128 does not prevent the loss of interrupts, you cannot use VDISK in extended memory.

Features

1. VDISK performs quickly because it operates at the speed of the computer's memory.
2. VDISK permits you to install more then one virtual disk in extended memory if you include the appropriate number of DEVICE=VDISK.SYS/E commands in your CONFIG.SYS file.
3. VDISK permits you to refer to each virtual disk with a letter, as you refer to disk drives.
4. You do not need to format virtual disks because DOS installs each disk in the formatted form.

CAUTION

> Each virtual disk you create increases the resident size of DOS by 720 bytes for the VDISK.SYS device driver plus the size of the buffer you specify, if the driver is installed in low memory.
>
> You lose the contents of a virtual disk if you reboot or lose the power on your computer.

Example

DEVICE=VDISK.SYS 160 256 32

This command installs a 160-KB virtual disk with 256-byte sectors and 32 directory entries.

Appendix E

How to Use Code Pages

INTRODUCTION

MS–DOS 3.3 provides national language support through the use of language–specific code pages. If you live in, or work with, a country other than the United States, you may choose to use the MS–DOS commands that support code page switching.

WHAT IS A CODE PAGE?

A code page is a table that defines the character set you are using. A character set is a country–specific or language– specific group of characters that are translated from the code page table and displayed by your screen or printer.

Each code page character set contains 256 characters. An example of a character set is the set of letters, numbers, and symbols (such as accent marks) used by French–Canadians.

MS–DOS 3.3 supports five different code pages:

437 – United States code page.
850 – Multilingual code page. This code page includes all characters for most languages of European, North American, and South American countries.
860 – Portuguese code page.
863 – French–Canadian code page.
865 – Nordic code page. This code page includes all characters for the Norwegian and Danish languages.

Tables E–1 through E–5 show the character sets for the five code pages supported.

Table E–1 Code Page 437 (United States) Character Set

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	NUL 00	► 16	32	0 48	@ 64	P 80	` 96	p 112	Ç 128	É 144	á 160	░ 176	└ 192	╨ 208	α 224	≡ 240
1	☺ 01	◄ 17	! 33	1 49	A 65	Q 81	a 97	q 113	ü 129	æ 145	í 161	▒ 177	┴ 193	╤ 209	β 225	± 241
2	☻ 02	↕ 18	" 34	2 50	B 66	R 82	b 98	r 114	é 130	Æ 146	ó 162	▓ 178	┬ 194	╥ 210	Γ 226	≥ 242
3	♥ 03	‼ 19	# 35	3 51	C 67	S 83	c 99	s 115	â 131	ô 147	ú 163	│ 179	├ 195	╙ 211	Π 227	≤ 243
4	♦ 04	¶ 20	$ 36	4 52	D 68	T 64	d 100	t 116	ä 132	ö 148	ñ 164	┤ 180	─ 196	╘ 212	Σ 228	⌠ 244
5	♣ 05	§ 21	% 37	5 53	E 69	U 85	e 101	u 117	à 133	ò 149	Ñ 165	╡ 181	┼ 197	╒ 213	σ 229	⌡ 245
6	♠ 06	▬ 22	& 38	6 54	F 70	V 86	f 102	v 118	å 134	û 150	ª 166	╢ 182	╞ 198	╓ 214	µ 230	÷ 246
7	• 07	↨ 23	' 39	7 55	G 71	W 87	g 103	w 119	ç 135	ù 151	º 167	╖ 183	╟ 199	╫ 215	τ 231	≈ 247
8	◘ 08	↑ 24	(40	8 56	H 72	X 88	h 104	x 120	ê 136	ÿ 152	¿ 168	╕ 184	╚ 200	╪ 216	Φ 232	° 248
9	○ 09	↓ 25	) 41	9 57	I 73	Y 89	i 105	y 121	ë 137	Ö 153	⌐ 169	╣ 185	╔ 201	┘ 217	Θ 233	∙ 249
A	◙ 10	→ 26	* 42	: 58	J 74	Z 90	j 106	z 122	è 138	Ü 154	¬ 170	║ 186	╩ 202	┌ 218	Ω 234	· 250
B	♂ 11	← 27	+ 43	; 59	K 75	[91	k 107	{ 123	ï 139	¢ 155	½ 171	╗ 187	╦ 203	█ 219	δ 235	√ 251
C	♀ 12	∟ 28	, 44	< 60	L 76	\ 92	l 108	¦ 124	î 140	£ 156	¼ 172	╝ 188	╠ 204	▄ 220	∞ 236	ⁿ 252
D	♪ 13	↔ 29	- 45	= 61	M 77	] 93	m 109	} 125	ì 141	¥ 157	¡ 173	╜ 189	═ 205	▌ 221	φ 237	² 253
E	♫ 14	▲ 30	. 46	> 62	N 78	^ 94	n 110	~ 126	Ä 142	₧ 158	« 174	╛ 190	╬ 203	▐ 222	ε 238	■ 254
F	☼ 15	▼ 31	/ 47	? 63	O 79	_ 95	o 111	⌂ 127	Å 143	ƒ 159	» 175	┐ 191	╧ 207	▀ 223	∩ 239	

Table E-2 Code Page 850 (Multilingual) Character Set

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	NUL 00	► 16	 32	0 48	@ 64	P 80	` 96	p 112	Ç 128	É 144	á 160	░ 176	└ 192	ð 208	Ó 224	- 240
1	☺ 01	◄ 17	! 33	1 49	A 65	Q 81	a 97	q 113	ü 129	æ 145	í 161	▒ 177	┴ 193	Đ 209	β 225	± 241
2	☻ 02	↕ 18	" 34	2 50	B 66	R 82	b 98	r 114	é 130	Æ 146	ó 162	▓ 178	┬ 194	Ê 210	Ô 226	= 242
3	♥ 03	‼ 19	# 35	3 51	C 67	S 83	c 99	s 115	â 131	ô 147	ú 163	│ 179	├ 195	Ë 211	Ò 227	¾ 243
4	♦ 04	¶ 20	$ 36	4 52	D 68	T 84	d 100	t 116	ä 132	ö 148	ñ 164	┤ 180	─ 196	È 212	õ 228	¶ 244
5	♣ 05	§ 21	% 37	5 53	E 89	U 85	e 101	u 117	à 133	ò 149	Ñ 165	Á 181	┼ 197	ı 213	Õ 229	§ 245
6	♠ 06	▬ 22	& 38	6 54	F 70	V 86	f 102	v 118	å 134	û 150	ª 166	Â 162	ã 198	Í 214	µ 230	÷ 246
7	• 07	↨ 23	' 39	7 55	G 71	W 87	g 103	w 119	ç 135	ù 151	º 167	À 183	Ã 199	Î 215	þ 231	¸ 247
8	◘ 08	↑ 24	(40	8 56	H 72	X 88	h 104	x 120	ê 136	ÿ 152	¿ 168	© 184	╚ 200	Ï 216	Þ 232	° 248
9	○ 09	↓ 25	) 41	9 57	I 73	Y 89	I 105	y 121	ë 137	Ö 153	⌐ 169	╣ 185	╔ 201	┘ 217	Ú 233	¨ 249
A	◙ 10	→ 26	* 42	: 58	J 74	Z 90	j 106	z 122	è 138	Ü 154	¬ 170	║ 186	╩ 202	┌ 218	Û 234	· 250
B	♂ 11	← 27	+ 43	; 59	K 75	[91	k 107	{ 123	ï 139	ø 155	½ 171	╗ 187	╦ 203	█ 219	Ù 235	¹ 251
C	♀ 12	∟ 28	, 44	< 60	L 76	\ 92	l 108	¦ 124	î 140	£ 156	¼ 172	╝ 188	╠ 204	▄ 220	ý 236	³ 252
D	♪ 13	↔ 28	- 45	= 81	M 77	] 93	m 109	} 125	ì 141	Ø 157	¡ 173	╜ 189	═ 205	¦ 221	Ý 237	² 253
E	♫ 14	▲ 30	. 46	> 62	N 78	^ 94	n 110	~ 126	Ä 142	× 158	« 174	¥ 190	╬ 206	Ì 222	· 238	■ 254
F	☼ 15	▼ 31	/ 47	? 63	O 79	_ 95	o 111	⌂ 127	Å 143	ƒ 159	» 175	┐ 191	╧ 207	▀ 223	´ 239	

Table E-3 Code Page 860 (Portuguese) Character Set

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	NUL 00	► 16	32	0 48	@ 64	P 80	` 96	p 112	Ç 128	É 144	á 160	░ 176	└ 192	╨ 208	α 224	≡ 240
1	☺ 01	◄ 17	! 33	1 49	A 65	Q 81	a 97	q 113	ü 129	À 145	í 161	▒ 177	┴ 193	╤ 209	β 225	± 241
2	☻ 02	↕ 18	" 34	2 50	B 66	R 82	b 98	r 114	é 130	È 146	ó 162	▓ 178	┬ 194	╥ 210	Γ 226	≥ 242
3	♥ 03	‼ 19	# 35	3 51	C 67	S 83	c 99	s 115	â 131	ô 147	ú 163	│ 179	├ 195	╙ 211	π 227	≤ 243
4	♦ 04	¶ 20	$ 36	4 52	D 68	T 84	d 100	t 116	ã 132	õ 148	ñ 164	┤ 180	─ 196	╘ 212	Σ 228	⌠ 244
5	♣ 05	§ 21	% 37	5 53	E 69	U 85	e 101	u 117	à 133	ò 149	Ñ 165	╡ 181	┼ 197	╒ 213	σ 229	⌡ 245
6	♠ 06	▬ 22	& 38	6 54	F 70	V 86	f 102	v 118	Á 134	Ú 150	ª 166	╢ 182	╞ 198	╓ 214	µ 230	÷ 246
7	• 07	↨ 23	' 39	7 55	G 71	W 87	g 103	w 119	ç 135	ù 151	º 167	╖ 183	╟ 199	╫ 215	τ 231	≈ 247
8	◘ 08	↑ 24	(40	8 56	H 72	X 88	h 104	x 120	ê 136	Ì 152	¿ 168	╕ 184	╚ 200	╪ 216	Φ 232	° 248
9	○ 09	↓ 25	) 41	9 57	I 73	Y 89	i 105	y 121	Ê 137	Ô 153	Ò 169	╣ 185	╔ 201	┘ 217	Θ 233	∙ 249
A	◙ 10	→ 26	* 42	: 58	J 74	Z 90	j 106	z 122	è 138	Ü 154	¬ 170	║ 186	╩ 202	┌ 218	Ω 234	· 250
B	♂ 11	← 27	+ 43	; 59	K 75	[91	k 107	{ 123	Ì 139	¢ 155	½ 171	╗ 187	╦ 203	█ 219	δ 235	√ 251
C	♀ 12	∟ 28	, 44	< 60	L 76	\ 92	l 108	¦ 124	Õ 140	£ 156	¼ 172	╝ 188	╠ 204	▄ 220	∞ 236	ⁿ 252
D	♪ 13	↔ 29	- 45	= 61	M 77	] 93	m 109	} 125	ì 141	Ù 157	¡ 173	╜ 189	═ 205	▌ 221	φ 237	² 253
E	♫ 14	▲ 30	. 46	> 62	N 78	^ 94	n 110	~ 126	Ã 142	₧ 158	« 174	╛ 190	╬ 206	▐ 222	ε 238	■ 254
F	☼ 15	▼ 31	/ 47	? 63	O 79	_ 95	o 111	⌂ 127	Â 143	Ó 159	» 175	┐ 191	╧ 207	▀ 223	∩ 239	

Table E–4 Code Page 863 (French–Canadian) Character Set

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	NUL 00	► 16	32	0 48	@ 64	P 80	` 96	p 112	Ç 128	É 144	¦ 160	░ 176	└ 192	╨ 208	α 224	≡ 240
1	☺ 01	◄ 17	! 33	1 49	A 65	Q 81	a 97	q 113	ü 129	È 145	´ 161	▒ 177	┴ 193	╤ 209	β 225	± 241
2	☻ 02	↕ 18	" 34	2 50	B 66	R 82	b 98	r 114	é 130	Ê 146	ó 162	▓ 178	┬ 194	╥ 210	Γ 226	≥ 242
3	♥ 03	‼ 19	# 35	3 51	C 67	S 83	c 99	s 115	â 131	ô 147	ú 163	│ 179	├ 195	╙ 211	Π 227	≤ 243
4	♦ 04	¶ 20	$ 36	4 52	D 68	T 84	d 100	t 116	Â 132	Ë 148	¨ 164	┤ 180	─ 196	╘ 212	Σ 228	⌠ 244
5	♣ 05	§ 21	% 37	5 53	E 69	U 85	e 101	u 117	à 133	Ï 149	¸ 165	╡ 181	┼ 197	╒ 213	σ 229	⌡ 245
6	♠ 06	▬ 22	& 38	6 54	F 70	V 86	f 102	v 118	¶ 134	û 150	³ 166	╢ 182	╞ 198	╓ 214	µ 230	÷ 246
7	• 07	↨ 23	' 39	7 55	G 71	W 87	g 103	w 119	ç 135	ù 151	¯ 167	╖ 183	╟ 199	╫ 215	τ 231	≈ 247
8	◘ 08	↑ 24	(40	8 56	H 72	X 88	h 104	x 120	ê 136	¤ 152	Î 168	╕ 184	╚ 200	╪ 216	Φ 232	° 248
9	○ 09	↓ 25	) 41	9 57	I 73	Y 89	i 105	y 121	ë 137	Ô 153	⌐ 169	╣ 185	╔ 201	┘ 217	Θ 233	∙ 249
A	◙ 10	→ 26	* 42	: 58	J 74	Z 90	j 106	z 122	è 138	Ü 154	¬ 170	║ 186	╩ 202	┌ 218	Ω 234	· 250
B	♂ 11	← 27	+ 43	; 59	K 75	[91	k 107	{ 123	ï 139	¢ 155	½ 171	╗ 187	╦ 203	█ 219	δ 235	√ 251
C	♀ 12	∟ 28	, 44	< 60	L 76	\ 92	l 108	¦ 124	î 140	£ 156	¼ 172	╝ 188	╠ 204	▄ 220	∞ 236	ⁿ 252
D	♪ 13	↔ 29	- 45	= 61	M 77	] 93	m 109	} 125	‗ 141	Ù 157	¾ 173	╜ 189	═ 205	▌ 221	φ 237	² 253
E	♫ 14	▲ 30	. 46	> 82	N 78	^ 94	n 110	~ 126	À 142	Û 158	« 174	╛ 190	╬ 206	▐ 222	ε 238	■ 254
F	☼ 15	▼ 31	/ 47	? 63	O 79	_ 95	o 111	⌂ 127	§ 143	ƒ 159	» 175	┐ 191	╧ 207	▀ 223	∩ 239	

Table E–5 Code Page 865 (Nordic) Character Set

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	NUL 00	► 16	32	0 48	@ 64	P 80	` 96	p 112	Ç 128	É 144	á 160	░ 176	└ 192	╨ 208	α 224	≡ 240
1	☺ 01	◄ 17	! 33	1 49	A 85	Q 81	a 97	q 113	ü 129	æ 145	í 161	▒ 177	┴ 193	╤ 209	β 225	± 241
2	☻ 02	↕ 18	" 34	2 50	B 66	R 82	b 98	r 114	é 130	Æ 146	ó 162	▓ 176	┬ 194	╥ 210	Γ 226	≥ 242
3	♥ 03	‼ 19	# 35	3 51	C 67	S 83	c 99	s 115	â 131	ô 147	ú 163	│ 179	├ 195	╙ 211	Π 227	≤ 243
4	♦ 04	¶ 20	$ 36	4 52	D 68	T 84	d 100	t 116	ä 132	ö 148	ñ 164	┤ 180	─ 196	╘ 212	Σ 228	⌠ 244
5	♣ 05	§ 21	% 37	5 53	E 69	U 85	e 101	u 117	à 133	ò 149	Ñ 185	╡ 181	┼ 197	╒ 213	σ 229	⌡ 245
6	♠ 06	▬ 22	& 38	6 54	F 70	V 86	f 102	v 118	å 134	û 150	ª 166	╢ 182	╞ 198	╓ 214	µ 230	÷ 246
7	• 07	↨ 23	' 39	7 55	G 71	W 87	g 103	w 119	ç 135	ù 151	º 167	╖ 183	╟ 199	╫ 215	τ 231	≈ 247
8	◘ 06	↑ 24	(40	8 56	H 72	X 88	h 104	x 120	ê 136	ÿ 152	¿ 166	╕ 184	╚ 200	╪ 216	Φ 232	° 248
9	○ 09	↓ 25	) 41	9 57	I 73	Y 89	i 105	y 121	ë 137	Ö 153	⌐ 169	╣ 185	╔ 201	┘ 217	Θ 233	∙ 249
A	◙ 10	→ 26	* 42	: 58	J 74	Z 90	j 103	z 122	è 138	Ü 154	¬ 170	║ 186	╩ 202	┌ 218	Ω 234	· 250
B	♂ 11	← 27	+ 43	; 59	K 75	[91	k 107	{ 123	ï 139	ø 155	½ 171	╗ 187	╦ 203	█ 219	δ 235	√ 251
C	♀ 12	∟ 28	, 44	< 60	L 76	\ 92	l 108	¦ 124	î 140	£ 156	¼ 172	╝ 188	╠ 204	▄ 220	∞ 236	ⁿ 252
D	♪ 13	↔ 29	- 45	= 61	M 77	] 93	m 109	} 125	ì 141	Ø 157	¡ 173	╜ 189	═ 205	▌ 221	φ 237	² 253
E	♫ 14	▲ 30	. 46	> 62	N 78	^ 94	n 110	~ 126	Ä 142	₧ 158	« 174	╛ 190	╬ 206	▐ 222	ε 238	■ 254
F	☼ 15	▼ 31	/ 47	? 63	O 79	_ 95	o 111	⌂ 127	Å 143	ƒ 159	¤ 175	┐ 191	╧ 207	▀ 223	∩ 239	

COUNTRY AND KEYBOARD CODES

MS-DOS also provides national language support through the use of two other codes:

> A country code defines the country in which you live or work. MS-DOS uses this code to prepare and assign default code pages for your system. MS-DOS recognizes 19 different country codes.
>
> A keyboard code defines the type of keyboard you are using. MS-DOS recognizes 17 different keyboard codes.

NATIONAL LANGUAGE SUPPORT CODES

The following table lists each country (or language) supported by MS-DOS 3.3. The table also lists the related country codes, default code page assignments, and related keyboard codes. The code pages shown are automatically prepared by MS-DOS when you load the corresponding country code through the CONFIG.SYS COUNTRY command. If you do not specify a country code, MS-DOS loads the default United States code page 437.

Country or Language	Country Code	Default Code Page Assignment	Keyboard Code
United States	001	437,850	US
French-Canadian	002	863,850	CF
Latin America	003	437,850	LA
Netherlands	031	437,850	NL
Belgium	032	437,850	BE
France	033	437,850	FR
Spain	034	437,850	SP
Italy	039	437,850	IT
Switzerland	041	437,850	SF,SG
United Kingdom	044	437,850	UK
Denmark	045	865,850	DK
Sweden	046	437,850	SV
Norway	047	865,850	NO
Germany	049	437,850	GR
English (International)	061	437,850	-
Portugal	351	860,850	PO
Finland	358	437,850	SU
Arabic countries	785	437	-
Israel	972	437	-

NOTE

Both Swiss-French and Swiss-German use country code 041. Code pages for Arabic and Hebrew languages are not available. Country codes 785 and 972 assume United States code page 437, but include country-specific date and time conventions.

COMMANDS THAT SUPPORT NATIONAL LANGUAGES

Several MS-DOS commands - new and old - support code page selection and national languages.

New MS-DOS Commands

MS-DOS 3.3 includes three new commands:

NLSFUNC Loads the file containing country-specific information.

CHCP Displays or changes the current code page for the system and all prepared devices.

Enhanced MS-DOS Commands

In addition to the new commands, MS-DOS 3.3 includes several enhanced MS-DOS commands that support code page selection. The most significant enhancements include

KEYB Allows you to select a country-specific keyboard code for the keyboard you are using, and a code page for the character set you prefer. You may also select an alternate keyboard definition file (other than the default keyboard.sys file) with this command, if another exists.

MODE Includes several new options

- Preparing a code page for a device
- Selecting a code page for a device
- Displaying the code pages prepared and selected for a device
- Refreshing code pages that were lost due to hardware error

SELECT Installs MS-DOS on a new floppy disk with selected country-specific information and keyboard code.

New and Enhanced Configuration Commands

Two CONFIG.SYS commands also support country-specific information:

COUNTRY — Identifies the country in which you work or live. This command also defines country-specific conventions to be used, such as date and time formats and sorting sequence for the character set.

DEVICE — Installs device drivers in the system, including two MS-DOS installable device drivers that support code page switching. These device drivers are called

DISPLAY.SYS - used to install a standard console screen device with code-page support

PRINTER.SYS - used to install a standard parallel printer with code-page support

Date and Time Formats

Four other MS-DOS commands - DATE, BACKUP, RESTORE, and TIME - now use country-specific date and time conventions, based on the code pages you choose to use.

The following table lists the date and time formats related to each country (or language group). These formats are determined by the country code set in your config.sys file.

For each country, the Date Format column shows how MS-DOS would display January 3, 1989, and the Time Format column shows how MS-DOS would display 5:35 p.m. (with zero seconds and zero hundredths of seconds).

Country or Language	Country Code	Date Format	Time Format
United States	001	1-03-1989	17:35:00.00
French-Canadian	002	1989-01-03	7:35:00.00
Latin America	003	03/01/1989	17:35:00.00
Netherlands	031	03-01-1989	17:35:00.00
Belgium	032	03/01/1989	17:35:00.00
France	033	03/01/1989	17:35:00.00
Spain	034	03/01/1989	17:35:00.00
Italy	039	03/01/1989	17:35:00.00
Switzerland	041	03.01.1989	17.35.00.00
United Kingdom	044	03-01-1989	17:35:00.00
Denmark	045	03/01/1989	17.35.00.00
Sweden	046	1989-01-03	17.35.00.00
Norway	047	03/01/1989	17.35.00.00
Germany	049	03.01.1989	17.35.00.00
English (International)	061	03-01-1989	17:35:00.00
Portugal	351	03/01/1989	17:35:00.00
Finland	358	03.01.1989	17.35.00.00
Arabic countries	785	03/01/1989	17:35:00.00
Israel	972	03 01 1989	17:35:00.00

HOW TO USE CODE PAGES

Setting the Code Page

Unless you specify otherwise, MS-DOS assumes that you want to use the United States character set. To set your system page to support another character set, you need to do four things:

1. Set the country code in your CONFIG.SYS file. This code identifies the country in which you live or work.
2. Load the COUNTRY.SYS file or other file containing the country-specific information for your country.
3. Set the system code page. For most country codes, MS-DOS automatically prepares two system code pages and selects the primary code page for your country automatically. If you want to use the other code page prepared for your country, you can use the CHCP command.
4. Set the keyboard code with the KEYB command.

NOTE

> Remember that when you change your CONFIG.SYS file, you must restart MS–DOS to enable the new settings.

Changing Character Sets

Example

Suppose you live in Quebec, Canada. You would follow these steps to use the French–Canadian character set with your system:

1. First, add the following line to your CONFIG.SYS file: COUNTRY=002
2. Then, restart MS–DOS so that MS-DOS can read the new setting.
3. Next, type the NLSFUNC command to load the country–specific information found in the COUNTRY.SYS file on your system:

   ```
   NLSFUNC
   ```

   NOTE

   > If you forget to type the NLSFUNC command, MS–DOS will not allow you to specify code pages or keyboard codes.

4. MS–DOS will automatically select the French–Canadian code page for you to use. Because your country code is 002, MS–DOS has also prepared the Multilingual code page for your system. If you would like to change the system code page, type CHCP 850.
5. Select the French–Canada keyboard code CF by typing the following command:

   ```
   KEYBCF
   ```

   NOTE

   > In place of steps 3, 4, and 5, you could add the following lines to your AUTOEXEC. BAT file. Then you would not have to type these commands each time you started MS–DOS.
   >
   > ```
   > NLSFUNC
   > CHCP 850
   > KEYBCF
   > ```

Now your computer is set up to use the French–Canadian character set. Since your console screen and printer are independent devices, you will also need to set them up for national language support. The next section explains how to do this.

HOW TO SET DEVICE CODE PAGES

MS–DOS 3.3 lets you define code pages for screen and parallel printer devices that support code pages switching. Unless you want to use the United States code page 437, you will want to set up your screen and printer to use the same code page as the rest of your system.

Setting Screen Code Pages

To set up your console screen device (CON) to use code pages, use the CONFIG.SYS device command to load the DISPLAY.SYS device driver.

Example

Suppose you are using an EGA display, and want to use the Multilingual code page 850. You could include this command in your CONFIG.SYS file:

```
    DEVICE=DISPLAY.SYS CON=(EGA,850,2)
```

The last option, 2, will allow you to prepare up to two code pages for this device. This is useful if you want to switch between code pages.

NOTE

> Remember to restart MS–DOS to initiate the changes you have made to the CONFIG.SYS file.

Setting Parallel Printer Code Pages

If you have a parallel printer connected to your personal computer, you will want to prepare the same code pages for your printer as for the rest of your system.

To do this, use the CONFIG.SYS device command again to load the installable device driver called PRINTER.SYS.

Example

If you have an IBM Proprinter, model 4201, connected to LPT1, you would include the following line in your CONFIG.SYS file:

```
    DEVICE=PRINTER.SYS LPT1=(4201,850,2)
```

This command line assumes that the PRINTER.SYS file is on the same disk as your CONFIG.SYS file. The last variable, 2, will let you prepare up to two code pages for this printer.

NOTE

> There is no limit to the number of times you can use the device command in your CONFIG.SYS file.

HOW TO SWITCH BETWEEN CODE PAGES

If you work in an environment that works with more than one language, you may need to switch between code pages. For example, suppose you work for an international company with offices in New York, London, Stockholm, and Oslo. You may need to use two or three different code pages to read or work from the correspondence you receive from your other offices.

Switching Between Code Pages

To illustrate how to switch code pages for your system and your devices, suppose that you want to change to Nordic code page 865 to work with some information you receive from the Oslo office. You would follow these steps:

1. First, be sure you have typed the NLSFUNC command. You only need to type this command once in order to load the country-specific information from the COUNTRY.SYS file.

2. Prepare the code page for each device you intend to use. For example, you would type the following command to prepare code page 865 for the parallel printer connected to LPT2:

   ```
   MODE LPT2 CODE PAGE PREPARE=865
   ```

   MS-DOS then displays the following message to let you know the code page was prepared for your device:

   ```
   MODE Prepare Code page function completed
   ```

   To prepare code page 865 for your console screen device (CON) you would type the following command:

   ```
   MODE CON CODE PAGE PREPARE=865
   ```

3. Next, change the code page for the system and all prepared devices by typing the following:

   ```
   CHCP 865
   ```

   The display on your screen may flicker slightly as MS-DOS loads a new code.

4. If for some reason you want to load a different code page for a single prepared device, you would use the select keyword with the MODE command.

   For example, to load code page 850 for your printer, type this command:

   ```
   MODE LPT2 CODE PAGE SELECT=865
   ```

   MS-DOS then displays the following message to let you know the code page was prepared for your device:

   ```
   MODE Select Code page function completed
   ```

   NOTE

   If you want to use these commands on a regular basis, you can include these command lines in your AUTOEXEC.BAT file.

HOW TO LIST CURRENT CODE PAGES

You can list the current prepared and selected code pages for your console screen or a parallel printer by using the MODE command in the following form:

```
MODE DEVICE CODE PAGE
```

For example, to display the current code pages for your console screen device, type the following:

```
MODE CON CODE PAGE
```

MS-DOS displays a message similar to this one:

```
Active code page for device CON is 437
          hardware code pages:
              Code page 850
          prepared codepages:
              Code page 437
              Code page 850
              Code page not prepared
              Code page not prepared
MODE Status Code page function completed
```

HOW TO REFRESH LOST CODE PAGES

It is possible for prepared code pages to be lost due to hardware errors or other reasons. For example, if you prepared code pages for your printer, and then turned off the printer, the current code page may be lost. You can use the refresh keyword with the MODE command to restore the lost code page.

To illustrate, suppose you had selected code page 850 as the active code page for your console screen (CON), but because of a hardware error, the active code page was lost. You could type the following commands to reinstate the active code pages for your screen:

```
MODE CON CODE PAGE PREPARE=((850) EGA.CPI)
MODE CON REFRESH
```

HOW TO FORMAT A DISK WITH COUNTRY-SPECIFIC INFORMATION

Using the SELECT Command

MS-DOS 3.3 includes a special command, SELECT, that will

- Format a disk
- Create a CONFIG.SYS file and AUTOEXEC.BAT files with country-specific information
- Copy the contents of the source disk to the target disk

CAUTION

> Do not use the SELECT command with a disk that already contains data files, unless you have backed up the files. Any data on the disk is destroyed when the disk is formatted by either the SELECT command or the FORMAT command.

Example

To illustrate how the SELECT command works, suppose after configuring your hard disk with FDISK you wanted to format your hard disk C. You also want to include the Latin American code page and keyboard code on your hard disk. After placing your MS-DOS master disk in drive A, you could type the following:

```
SELECT A: C:003 LA
```

After formatting the disk in drive C, SELECT creates two files on the target disk - AUTOEXEC.BAT and CONFIG.SYS. The contents of the AUTOEXEC.BAT file will look something like this:

```
PATH C:;
KEYBLA 437
ECHO OFF
CLS
DATE
TIME
VER
```

The contents of the CONFIG.SYS file will look similar to the following:

```
COUNTRY=003,437
```

Finally, the SELECT command copies the MS-DOS files to the disk on drive D. If AUTOEXEC.BAT and CONFIG.SYS files exist on drive A, SELECT does not copy them to drive C.

Appendix F

Disk and Device Errors

This appendix describes MS–DOS disk and device error messages. Refer to Appendix G, "MS–DOS Message Directory," for other MS–DOS messages.

If a disk or device error occurs at any time during a command or program, MS–DOS displays an error message, indicates whether it occurred while reading from or writing to the device, and lists the name of the device (such as PRN or drive A) that has the error.

When MS–DOS displays an error message it waits for you to type one of the following responses:

A Abort. End the program requesting the disk read or write.

I Ignore. Ignore the bad sector and pretend the error did not occur. This may result in lost data.

R Retry. Repeat the operation. You should use this response when you have corrected the error (for example, with "Not ready" or "Write protect" errors).

F Fail. This causes the current MS–DOS system operation to end (fail) and the application to continue.

Usually, you will want to recover by typing responses in this order:

R (to try again)

A (to terminate a program and try a new disk)

ERROR MESSAGES

Bad call format error

The length of the request header passed to the device header was incorrect.

Bad command error

A device driver issued an incorrect command to the device specified in the error message.

Bad unit error

Invalid subunit numbers were passed to the device driver.

Data error

MS-DOS could not read the data from the disk properly. This is often due to a defective disk. Try choosing **R** (for Retry) several times, or choose **A** (for Abort) to end the program. (It's a good idea to make a new copy of the disk, because if it's defective, you may lose information.)

FCB unavailable General failure error

An unusual error has occurred. This error usually requires an experienced programmer to fix it. Choose **R** (for Retry) or **A** (for Abort).

File Allocation Table bad for drive *x*

This message means that the copy in memory of one of the allocation tables points to nonexistent blocks. Possibly the disk was incorrectly formatted or not formatted before use. If this error persists after you format it, the disk is unusable.

Invalid disk change error

You changed the disk in a drive when you weren't supposed to. Put the disk back in the drive and choose **R** (for Retry).

Lock violation error

A program tried to access part of a file that another program was using. Choose **A** (for Abort), or wait awhile and choose **R** (for Retry).

No paper error

The printer is either out of paper or not turned on.

Non-DOS disk error

MS-DOS does not recognize the disk format because the disk is missing information or contains another operating system. Try running the chkdsk command to correct the problem. (See Section 4, "MS-DOS Commands," for information about CHKDSK.) If running CHKDSK does not solve the problem, you should reformat the disk by using the FORMAT command--even though this will destroy all the files on the disk.

Not ready error

The device (usually a drive or printer) specified in the error message is not ready to accept or transmit data. This often happens when the disk drive door is open. If this is the problem, close the door and choose **R** (for Retry), or check to see if the printer is on and ready.

Read fault error

MS-DOS is unable to read data from the device (usually a disk drive). Check to see that the disk is properly inserted in the drive, then choose **R** (for Retry).

Sector not found error

This error usually means the disk has a defective spot so that MS-DOS cannot find the requested information on it. You should copy all files from the disk onto a good disk and then try to reformat the defective disk.

Seek error

MS-DOS is unable to locate the information on the disk. Make sure that the disk is properly inserted in the drive, or try a different drive.

Sharing violation

A program tried to access a file that another program was currently using. Choose A (for Abort), or wait awhile and choose **R** (for Retry).

Write fault error

MS-DOS is unable to write data to the specified device. Make sure that the disk is properly inserted in the disk drive. Try **R** (for Retry). If the error occurs again, choose **A** (for Abort).

Write protect error

You tried to write data on a write-protected disk. If the disk has a write-protect tab on it, you must remove the tab before you can write on the disk. (You should consider first why the disk was write-protected.) If the disk doesn't have a write-protect notch, you cannot write on that disk. a write-protect notch, you cannot write on that disk.

Appendix G

MS-DOS Message Directory

This section describes MS-DOS messages, their causes, and how to correct them. It includes a listing of the commands in brackets ([ ]) that cause each message.

For information specifically on device error messages, see Appendix F, "Disk and Device Errors."

Abort edit (Y/N)?
[EDLIN]

MS-DOS displays this message when you choose the EDLIN Q (QUIT) command. The Q command exits the editing session without saving any editing changes. Type **Y** (for Yes) or **N** (for No).

Abort, Retry, Ignore?
[MS-DOS]

This is a device error. See Appendix F, "Disk and Device Errors."

Access denied
[ATTRIB][FIND][PRINT][REPLACE][XCOPY]

You tried to replace a write-protected, read-only, or locked file.

Active Code Page: xxx
[CHCP]

xxx is the code page currently being used by the system.

Active Code Page for device ddd is xxx
[MODE]

xxx is the code page currently being used by the device ddd.

Active Code Page not available from CON device
[KEYB]

The code page that the system is currently using is not supported on the console (screen) you are using.

Add filename?
(Y/N) [REPLACE]

REPLACE displays this prompt if you specify the /W switch. Press **Y** if you want to add the file to the disk, or **N** if you do not want to add the file.

Adding filename
[REPLACE]

Replace displays this prompt to let you know that it is adding this file to your disk.

All files canceled by operator
[PRINT]

MS-DOS displays this message when you specify the /T switch with the PRINT command.

All logical drives deleted in the Extended DOS Partition
[FDISK]

Any logical drives previously associated with the extended DOS partition on your disk are now removed.

Allocation error, size adjusted
[CHKDSK]

The size of the file indicated in the directory was not consistent with the amount of data actually allocated to the file. The file was truncated to match the amount of data allocated.

All specified file(s) are contiguous
[CHKDSK]

All files are written sequentially on the disk. To correct this error automatically, specify the CHKDSK /F switch.

APPEND already installed
[APPEND]

You have already used the APPEND command once since you turned on your computer. Now you are trying to use either the /x or /e switch with this command. These switches are only valid the first time you type the APPEND command.

If you want to change the APPEND switch, reboot your computer. Then type the APPEND command with the switch you want to use. Otherwise, use the APPEND command without these switches. For more information about the APPEND command, see Section 4, "MS-DOS Commands."

APPEND/ASSIGN Conflict
[APPEND]

You cannot use the APPEND command on an assigned drive. Cancel the drive assignment before using the APPEND command with this drive again.

Are you sure (Y,N)?
[MS-DOS]

MS-DOS displays this message if you try to delete all files in the working directory by using the *.* wildcard. Type **Y** (for Yes) to delete all the files, or **N** (for No).

Attempted write-protect violation
[FORMAT]

The disk you are trying to format is write-protected.

***** Backing up files to drive *x*: *****
Diskette Number: *n*
[BACKUP]

BACKUP displays this message while backing up files to the specified target drive. Be sure to label backup disks with the appropriate backup disk number for use in restoring them later.

Bad call format reading (or writing) drive *x*:
[MS-DOS]

This is a device error. See Appendix F, "Disk and Device Errors."

Bad command error reading (or writing) drive *x*:
[MS–DOS]

This is a device error. See Appendix F, "Disk and Device Errors." Bad command or file name [MS–DOS]

The command cannot find the program you asked it to run. Check to see that you typed the command line properly, and that the file or command is on the disk, or in the command path.

Bad command or file name
[MS–DOS]

The command cannot find the program, you asked it to run. Check to see that you typed the command line properly, and that the file or command is on the disk, or in the command path.

Bad or missing Command Interpreter
[MS–DOS]

MS–DOS cannot find the COMMAND.COM file on the disk; either the file is missing from the root directory, or the file is invalid. You also receive this message if COMMAND.COM has been moved from the directory it was originally in when you started MS–DOS. Either restart the system with a disk that contains the COMMAND.COM file, or copy the COMMAND.COM file from your backup MS–DOS master disk onto the disk used to start MS–DOS.

Bad or missing filename
[MS–DOS]

You specified a device incorrectly in the CONFIG.SYS file. Check the accuracy of the device command in the CONFIG.SYS file.

Bad or Missing Keyboard definition file
[KEYB]

MS–DOS cannot find the Keybxx file that you specified with the KEYB command. Check to see that the file you specified exists on the disk. Also check to see that your path includes the directory in which this file resides. Then, retype the command. If you get this message again, the KEYBOARD.SYS or KEYB.COM file may be corrupted.

Bad Partition Table
[FORMAT]

This message means that there is no DOS partition on the hard disk. You must run fdisk to create a DOS partition on your hard disk.

Bad unit error reading drive ***x*****:**
[MS–DOS]

This is a device error. See Appendix E, "Disk and Device Errors."

BREAK is off (or on)
[MS–DOS]

This message tells you the current setting of BREAK.

Cannot CHDIR to path tree past this point not processed
[CHKDSK]

CHKDSK is checking the structure of the directory and is unable to go to the specified directory. All subdirectories underneath this directory will not be verified. To correct this error automatically, specify the CHKDSK /F switch. Cannot CHDIR to root [CHKDSK]

CHKDSK is checking the tree structure of the directory and is unable to return to the root directory. CHKDSK is not able to continue checking the remaining subdirectories. Try to restart MS–DOS. If this error persists, the disk is unusable.

Cannot CHKDSK a Network drive
[CHKDSK]

You cannot check drives that are redirected over the network.

Cannot CHKDSK a SUBSTed or ASSIGNed drive
[CHKDSK]

You cannot check drives that have been substituted or assigned.

Cannot COPY from (or to) a reserved device
[XCOPY]

You cannot copy files from or to a device.

Cannot create extended DOS partition while logical drives exist
[FDISK]

Your disk has one or more logical drives assigned to it. These must be deleted before you can create an extended DOS partition. Delete all logical drives by using FDISK. Then create the extended DOS partition.

Cannot create extended DOS partition without primary DOS partition on disk 1
[FDISK]

You are trying to create an extended DOS partition, but your first hard disk does not contain a primary DOS partition. First, create the primary DOS partition on your first hard disk. Then, if you have more room on that disk, or if you have a second hard disk, you can create an extended DOS partition.

Cannot create Subdirectory BACKUP on drive *x*:
[BACKUP]

The disk may be write-protected, full, or the backup subdirectory may already exist and be read-only. Use another disk as a target disk.

Cannot create a zero cylinder partition
[FDISK]

You are trying to create a partition with a size of 0 cylinders. You must allocate a minimum of 1 cylinder to any partition you create.

Cannot DISKCOMP to or from an ASSIGNed or SUBSTed drive
[DISKCOMP]

One of the drives that you specified is a drive that you created using the ASSIGN or SUBST command.

Cannot DISKCOMP to or from a network drive
[DISKCOMP]

You cannot compare disks on drives that have been redirected over the network.

Cannot DISKCOPY to or from an ASSIGNed or SUBSTed drive
[DISKCOPY]

One of the specified drives was created using the ASSIGN or SUBST command.

Cannot DISKCOPY to or from a network drive
[DISKCOPY]

You cannot copy disks to or from drives that have been redirected over the network.

Cannot do binary reads from a device
[COPY]

The copy cannot be done in binary mode when you are copying from a device. You should either not use the /B switch, or you should use the /A switch to specify an ASCII copy.

Cannot edit .BAK file--rename file
[EDLIN]

You attempted to edit a file that had a filename extension of .BAK (a backup copy created by EDLIN). If you must edit a file that has an extension of .BAK, you must either rename or copy the file and give it a different extension.

Cannot exec BASICA.COM
[MS–DOS]

BASICA cannot be executed by MS–DOS. Check to see that the basica.com file is on the disk you are using. If BASICA.COM is not in your working directory, make sure that the PATH command points to the directory in which it is located. Try executing BASICA again. If you get the same message, the file itself may be bad. Try restoring BASICA.COM from backup.

Cannot format an ASSIGNed or SUBSTed drive
[FORMAT]

You attempted to format a drive currently mapped to another drive by the ASSIGN or SUBST command. Run assign or subst again and clear all drive assignments.

Cannot FORMAT a Network drive
[FORMAT]

You cannot format drives that are redirected over the network.

Cannot FORMAT nonremovable drive x
[BACKUP]

You are trying to back up files with the /f switch. MS-DOS will not allow you to format the target disk specified. Be sure you want to back up files to a hard disk. If you do, you must use a hard disk that is formatted already.

Cannot JOIN a Network drive
[JOIN]

You cannot join drives that are redirected over the network.

Cannot LABEL a Network drive
[LABEL]

You cannot label a drive that is shared on a network server station.

Cannot LABEL a SUBSTed or ASSIGNed drive
[LABEL]

You cannot label a drive if it has been substituted with the SUBST command or assigned with the ASSIGN command. Check the command line to be sure you specified a valid filename.

Cannot perform a cyclic copy
[XCOPY]

When you are using the /S switch, you may not specify a target that is a subdirectory of the source.

Cannot recover . entry, processing continued
[CHKDSK]

The "." entry (working directory) is defective and can not be recovered.

Cannot recover .. entry, Entry has a bad attribute (or link or size)
[CHKDSK]

The ".." entry (parent directory) is defective and cannot be recovered. If you have specified the /F switch, CHKDSK tries to correct the error automatically.

Cannot RECOVER a Network drive
[RECOVER]

You cannot recover files on drives that are redirected over the network.

Cannot SUBST a Network drive
[SUBST]

You cannot substitute drives that are redirected over the network.

Cannot SYS to a Network drive
[SYS]

You cannot transfer the system files to drives that are redirected over the network. For more information about the NET PRINT command, see the *Microsoft Networks User's Guide*.

Cannot use FASTOPEN for drive x:
[FASTOPEN]

FASTOPEN works only with local, fixed disks and can work with a maximum of four disks at a time. You may be trying to use FASTOPEN over a network, with a floppy disk, or with more than four disks at one time, none of which is possible with FASTOPEN.

Cannot use PRINT – Use NET PRINT
[PRINT]

You must use the NET PRINT command to print files.

CHDIR .. failed, trying alternate method
[CHKDSK]

When checking the tree structure, CHKDSK was not able to return to a parent directory. It will try to return to that directory by starting over at the root and searching again.

Code page not prepared
[MODE]

You have selected a code page that has not yet been prepared for the system, or one that does not have the correct font to support the current video mode.

To prepare a code page for the system, use the MODE PREPARE command. If you have installed the DISPLAY.SYS installable device driver, be sure the DEVICE command line in your CONFIG.SYS file allows for additional subfonts. For more information, see Appendix D, "Installable Device Drivers", and Appendix C, "How to Configure Your System."

Code page xxx not prepared for all devices
[CHCP]

You have selected a code page that is not currently supported by a device. First, be sure your device supports code page switching, and that it is currently on-line. If the device supports code page switching, use the MODE PREPARE command to prepare the device for the code page. Then retry the CHCP command.

Code page xxx not prepared for system
[CHCP]

CHCP is unable to select a code page for the system.

First, make sure that NLSFUNC is installed. If you have not used the DEVICE command in your CONFIG.SYS file to install device drivers, you may now retry the CHCP command. If you are using installable device drivers with your system, you must use the MODE PREPARE command to prepare the specific code page for each device on your system. Then retype the CHCP command.

Code page operation not supported on this device
[MODE]

You have specified a device and code page combination which MS-DOS does not recognize as valid. Check to see that the device you specified exists and that you have listed a valid code page. Also check to see that that code page is supported on that device.

Code page requested xxx is not valid for given keyboard code
[KEYB]

The keyboard code and code page specified are not compatible. Retype the KEYB command with compatible keyboard code and code page.

Code page specified has not been designated
[KEYB]

You have typed the KEYB command with an option the system does not recognize. You must first prepare the associated code page for your console screen device.

Use the MODE PREPARE command to prepare the associated code page for CON. Then retype the KEYB command.

Code page specified has not been prepared
[KEYB]

You have typed the KEYB command with an option the system does not recognize. You must first prepare the associated code page for your console screen device. Use the MODE PREPARE command to prepare the associated code page for CON. Then retype the KEYB command.

Code page specified is inconsistent with invoked code page
[KEYB]

This warning message lets you know that the KEYB option you've selected does not coincide with the code page for your console screen device (CON). Use the MODE SELECT command if you also want to change the code page for CON.

Code page specified is inconsistent with selected code page
[KEYB]

This warning message lets you know that the KEYB option you've selected does not coincide with the code page for your console screen device (CON). Use the MODE SELECT command if you also want to change the code page for CON.

Code pages cannot be prepared
[MODE]

You have either specified a duplicate code page for this device or tried to prepare more than the total number of code pages supported for this device. Check the DEVICE command line in your CONFIG.SYS file to see how many prepared code pages are allowed for this device. Use the /status option of the MODE command to find out which code pages are already prepared for this device. For more information, see Appendix C, "How to Configure Your System" and Section 4, "MS–DOS Commands."

Compare another diskette (Y/N)?
[DISKCOMP]

DISKCOMP displays this message when it has completed its comparison of the disks. Press Y (for Yes) if you want to compare more disks.

Compare error on disk
side *s*, track *t*
[DISKCOMP]

DISKCOMP found a difference on the disk in the specified drive, side *s*, track *t*.

Compare OK
[DISKCOMP]

DISKCOMP displays this message if the disks are identical.

Compare process ended
[DISKCOMP]

DISKCOMP displays this message if a fatal error occurred during the comparison.

Comparing t tracks
n sectors per track, s side(s)
[DISKCOMP]

This message confirms the format of the disks that you are comparing.

COM port does not exist
[MODE]

You have specified an invalid COM port.

Contains *n* non–contiguous blocks.
[CHKDSK]

The disk contains fragmented files. If you want to copy this disk, you should use the COPY or XCOPY command instead of the DISKCOPY command. The new copy will then store the new files sequentially.

Content of destination lost before copy [COPY]

The source file that you specified in the COPY command was overwritten before the copy process completed. Refer to the COPY command for the proper syntax.

Convert lost chains to files (Y/N)?
[CHKDSK]

CHKDSK displays this message if it finds information on the disk that isn't allocated properly in the disk's File Allocation Table. If you type **Y** (for Yes) in response to this prompt, CHKDSK recovers the lost blocks it found when check ing the disk. CHKDSK then creates a proper directory entry and a file for each lost chain with a filename of the form: file*nnnn*.CHK. If you type **N** (for No), CHKDSK frees the lost blocks so that they can be reallocated and does not recover any data that was in those lost blocks.

Copy another diskette (Y/N)?
[DISKCOPY]

The DISKCOPY command has completed processing. Type **Y** (for Yes) if you want to copy another disk, or **N** (for No) if you don't.

Copying *t* tracks
***n* Sectors/Track, *s* Sides**
[DISKCOPY]

DISKCOPY displays this message during copying.

Copy process ended
[DISKCOPY]

DISKCOPY could not copy the entire disk. Use the COPY or XCOPY command to copy specific files onto the disk.

Copyright 1981,82,83,84,85,86 Microsoft Corp.
[MS–DOS]

This message appears on most MS–DOS utility and command banners.

Corrections will not be written to disk
[CHKDSK]

There are errors on the disk, but CHKDSK will not correct them because you did not specify the /F switch. You must specify the CHKDSK switch to correct disk errors.

Current code page settings
[MODE]

This informational message shows current and prepared code pages for the device specified and for the system.

Current date is ***mm-dd-yy***
[DATE]

The DATE command displays this message. Enter the correct date and press **Enter**.

Current keyboard code: xx code page: yyy Current CON code page: zzz
[KEYB]

This message displays the current keyboard code and its associated code page, and current code page used by your console screen device (CON).

Current keyboard does not support this code page
[KEYB]

The code page selected is not compatible with the current keyboard code. Check the code page you have selected. If it is correct, change the keyboard code with the KEYB command.

Current time is ***hh:mm:ss.hh***
[TIME]

The TIME command displays this message. Enter the correct time and press **Enter**.

Data error reading drive ***x*****:**
[MS-DOS]

This is a device error. See Appendix F, "Disk and Device Errors."

Delete current volume label (Y/N)?
[LABEL]

If a current volume label exists, Label displays this message in response to the prompt to enter the new volume label for this disk. If you want to delete the volume label, press **Y** (for Yes); otherwise, press **N** (for No).

xxxxxxxx **device driver cannot be initialized**
[MS-DOS]

You are trying to install a device driver by using a DEVICE command line in your CONFIG.SYS file. The syntax on that command line is wrong. See Appendix D, "Installable Device Drivers", for the correct syntax of MS-DOS installable device drivers.

Device Error during Status
[MODE]

MS–DOS found an error with the specified device when it was checking the status of that device. The problem may be due to a device that does not support code pages, a device not properly prepared for code page switching, a device which cannot support more code pages than those already prepared, or a device with a bad or irregular font file. Check the DEVICE command line in your CONFIG.SYS file. Make sure that the command syntax and limits for subfonts and additional code pages are all correct. Also check to see if your device supports code page switching. Consult the hardware vendor if you are unsure.

Device Error during Prepare
[MODE]

MS–DOS found an error with the specified device when preparing that device for code page switching. The problem may be due to a device that does not support code pages, a device not properly prepared for code page switching, a device which cannot support more code pages than those already prepared, or a device with a bad or irregular font file. Check the DEVICE command line in your CONFIG.SYS file. Make sure that the command syntax and limits for subfonts and additional code pages are all correct. Also check to see if your device supports code page switching. Consult the hardware vendor if you are unsure.

Device Error during Select
[MODE]

MS–DOS found an error with the specified device. The problem may be due to a device that does not support code pages, a device not properly prepared for code page switching, a device which cannot support more code pages than those already prepared, or a device with a bad or irregular font file. Check the DEVICE command line in your CONFIG.SYS file. Make sure that the command syntax and limits for subfonts and additional code pages are all correct. Also check to see if your device supports code page switching. Consult the hardware vendor if you are unsure.

Device Error during write of font file to device
[Mode]

MS–DOS found an error when it tried to write the font file to the specified device. The problem may be due to a device that does not support code pages, a device not properly prepared for code page switching, a device which cannot support more code pages than those already prepared, or a device with a bad or irregular font file. Check the DEVICE command line in your CONFIG.SYS file. Make sure that the command syntax and limits for subfonts and additional code pages are all correct. Also check to see if your device supports code page switching. Consult the hardware vendor if you are unsure.

Device or code page missing from font file
[MODE]

MS-DOS did not find a definition of the indicated code page for this device in the font file. Use the MODE command to specify another code page for this device. Also check to see that the font file supports the code page you want to use. This error also may cause specified code pages to be undefined. Use the MODE command to prepare and refresh lost code pages.

Device ddd not prepared
[MODE]

No code page has been prepared for this device.

DEVICE Support Not Present
[DISKCOMP][DISKCOPY]

The disk drive does not support MS-DOS 4.0 device control.

Directory is joined
[CHKDSK]

CHKDSK does not process directories that are joined. Use the JOIN /D command to "unjoin" the directories, and then run CHKDSK again.

Directory is totally empty,
no . or ..
[CHKDSK]

The specified directory does not contain references to working and parent directories. Delete the specified directory and re-create it.

Directory not empty
[JOIN]

You can only join onto an empty directory.

Disk error reading (or writing) drive *x*:
[MS-DOS]

This is a device error. See Appendix F, "Disk and Device Errors."

Disk error reading (or writing) FAT
[CHKDSK]

One of your File Allocation Tables has a defective sector in it. MS–DOS automatically uses the other FAT. You should copy all your files onto another disk. To correct this error automatically, you simply specify the CHKDSK /F switch.

Diskette bad or incompatible
[DISKCOPY]

The source disk is not formatted, or was formatted incorrectly. You cannot copy it.

Disk full. Edits lost
[EDLIN]

EDLIN was not able to save your file due to lack of disk space. You should always make sure that there is enough room on the default disk to save your file before you give the EDLIN E (END) command. You should also make sure that the default disk is not write–protected.

Disk unsuitable for system disk
[FORMAT]

The FORMAT program detected a bad track on the disk where system files should reside. You should use this disk to store data only.

Does *name* specify a file name
or directory name on the target
(F = file D = directory)?
[XCOPY]

XCOPY displays this prompt if the target directory does not exist. Type **F** if the name specifies a file, or **D** if the target specifies a directory that does not currently exist.

(.)(..) **Does not exist**
[CHKDSK]

This is an informational message from CHKDSK, indicating that either the "." or ".." directory entry is invalid.

Do not specify filename(s)
Command format: DISKCOMP d: d:[/1][/8]
[DISKCOMP]

You specified an incorrect switch or gave a filename in addition to a drive name.

Do not specify filename(s)
Command format: DISKCOPY d: d:[/1]
[DISKCOPY]

You specified an incorrect switch or gave a filename in addition to a drive name.

DOS 2.0 or later required
[ATTRIB][BACKUP][FC][GRAPHICS][JOIN][MODE]
[RESTORE][SUBST]

You cannot use these utilities with 1.xx versions of MS-DOS.

Do you see the leftmost 0? (Y/N)
[MODE]

MODE displays this message to help you align the test pattern on your screen. Type **Y** (for Yes) if you can see the leftmost 0 in the test pattern, or type **N** (for No) if you want to shift the display to the right.

Do you see the rightmost 9? (Y/N)
[MODE]

MODE displays this message to help you align the test pattern on your screen. Type **Y** (for Yes) if you can see the rightmost 9 in the test pattern, or type **N** (for No) if you want to shift the display to the left.

Do you wish to use the maximum size for a DOS partition and make the DOS partition active (Y/N)........[]
[FDISK]

You are formatting your hard disk. Type **Y** (for Yes) and press the **ENTER** key if you want to format your entire hard disk as the primary DOS partition. Otherwise, type **N** (for No) and press the **ENTER** key.

Drive D already deleted
[FDISK]

You tried to delete drive D, but it had already been deleted.

Drive deleted
[FDISK]

You deleted a hard drive from the system.

Drive has been changed or deleted
[FDISK]

You changed or deleted a hard drive on the system.

Drive letter must be specified
[FORMAT]

You did not specify the drive letter for the drive that you want to format. You must specify the name of the drive that you want to format.

Drive *x*: not ready
Make sure a diskette is inserted into
the drive and the door is closed
[DISKCOMP] [DISKCOPY]

The drive is empty, or you did not close the door of the disk drive.

Drive types or diskette types not compatible
[DISKCOMP] [DISKCOPY]

You must have the same size and type of disks to run these commands. For example, you cannot copy from a single–sided disk to a double–sided disk, or compare a high–density disk with a low–density disk. You should use FC if you want to compare the files on the disks. If you want to copy the disk, you can use COPY or XCOPY, or reformat the target disk so that it's the same type as the source disk, or use a disk of the same type.

Duplicate file name
[RENAME]

You tried to rename a file to a filename that already exists, or the name you specified could not be found.

ECHO is off (or on)
[MS–DOS]

This message tells you the current status of ECHO.

End of input file
[EDLIN]

The entire file was read into memory. If the file was read in sections, thiѕ message indicates that the last section of the file is in memory.

Enter current Volume Label for drive *x*:
[FORMAT]

FORMAT asks you to enter the current volume label for verification before it formats the hard disk in the specified drive. If you do not know what the volume label is, press **Ctrl–C** to abort this command, and give the VOL command for the specified drive. Then give the FORMAT command again.

Enter new date:
[DATE]

You must respond to this prompt when you start MS–DOS, or when you use the DATE command. Enter the date in a *mm–dd–yy* format, or press the **Enter** key to accept the current date.

Enter new time:
[TIME]

You must respond to this prompt when you start MS–DOS. Enter the time in the *hh:mm* format, or press the **Enter** key to accept the current time.

Entry error
[EDLIN]

The last command you typed contained a syntax error. Retype the command with the correct syntax and press the **Enter** key.

Entry has a bad attribute (or link or size)
[CHKDSK]

This message may be preceded by one or two periods that show which subdirectory is invalid. If you have specified the /F switch, CHKDSK tries to correct the error automatically.

Error during read of Font file
[MODE]

MS–DOS found an error when it tried to read the font file for the code page specified.

Error in COUNTRY command
[MS–DOS]

You used the incorrect syntax for the COUNTRY command in your CONFIG.SYS file. For the correct syntax of this configuration command, see Appendix C, How to Configure Your System.

Error in .EXE file
[MS–DOS]

The .EXE file you have asked MS–DOS to load has an invalid internal format. You cannot run this program. Check to make sure that you are using the correct version of MS–DOS.

Error reading/writing partition table
[FORMAT]

FORMAT could not read or write the partition table. You should run FDISK on the disk and then try formatting it again.

Errors found, F parameter not specified
Corrections will not be written to disk
[CHKDSK]

CHKDSK found errors on the disk. If you have not specified the /F switch, CHKDSK continues printing messages but will not correct the errors. You should run CHKDSK with the /F switch if you want to correct the problems encountered by the CHKDSK command.

Errors on list device indicate that it
may be off–line. Please check it.
[PRINT]

Your printer is not turned on.

Error trying to open backup log file
Continuing without making log entries.
[BACKUP]

You specified the BACKUP /L switch, but BACKUP could not create the backup log file.

Error writing to device
[MS–DOS]

You tried to send too much data to a device, so MS–DOS was unable to write the data to that device.

EXEC failure
[MS–DOS]

MS–DOS either found an error when reading a command, or the FILES command in the CONFIG.SYS file is set too low. Increase the value of the files command in the CONFIG.SYS file, and restart MS–DOS.

Extended DOS partition already exists
[FDISK]

You cannot create another extended DOS partition.

Extended DOS partition created
[FDISK]

You have created an extended DOS partition on your hard disk.

Extended DOS partition deleted
[FDISK]

You have deleted an extended DOS partition from your hard disk.

Failure to access code page font file
[MODE]

MS–DOS cannot open the font file for the specified code page. Check to see that you typed font file name, and its pathname correctly. Also check the CONFIG.SYS file to see that the device driver for this device has been properly installed. If the CONFIG.SYS file is incorrect, correct it and restart MS–DOS before retyping the MODE command.

Failure to access COUNTRY.SYS
[SELECT]

MS–DOS cannot open the COUNTRY.SYS file. Check to see that your path points to the directory in which COUNTRY.SYS resides. Then retype the command.

Failure to access device: xxx
[MODE]

You are trying to specify a code page for a particular device, but MS–DOS cannot access the device listed. Retype the command using an existing device. Make sure you are typing the device name correctly.

Failure to access KEYBOARD.SYS
[SELECT]

MS–DOS cannot open the KEYBOARD.SYS file. Be sure KEYBOARD.SYS exists on your source disk.

FASTOPEN already installed
[FASTOPEN]

FASTOPEN is already installed on the system.

FASTOPEN installed
[FASTOPEN]

This informational message acknowledges that you have installed FASTOPEN.

FCB unavailable reading (or writing) drive *x*:
[MS–DOS]

This is a device error. See Appendix F, "Disk and Device Errors."

fc: cannot open *filename* – No such file or directory
[FC]

One of the files that you specified doesn't exist. Check the directory for the correct filename.

fc: *filename* longer than *filename*
[FC]

After reaching the end of one of the files in a file comparison, the other file still has uncompared data left.

fc: incompatible switches
[FC]

You have specified switches that are not compatible. (For example, /B and /L.) You should not combine binary and ASCII comparison switches.

fc: no differences encountered
[FC]

The files are the same.

fc: out of memory
[FC]

You do not have enough memory to perform the comparison.

File allocation table bad
[MS-DOS]

The disk may be defective. Run CHKDSK /F to check the disk.

File allocation table bad drive *x*:
[CHKDSK]

This message means that the disk was not formatted or was formatted improperly. It could also mean that an operating system other than MS-DOS is on the disk. Run CHKDSK /F to check the disk. If this message is displayed again, you must reformat the disk.

File *filename* canceled by operator
[PRINT]

MS-DOS displays this message when you specify the /T switch with the PRINT command.

File cannot be converted
[EXE2BIN]

The input file is not in the correct format.

File cannot be copied onto itself
[COPY] [REPLACE] [XCOPY]

The source filename you specified is the same as the target filename.

File creation error
[MS-DOS] [EDLIN] [RESTORE] [XCOPY]

You tried to add a new filename or replace a file that already exists in the directory, or there was not enough space for the file. If the file already exists, it is a read only file and cannot be replaced. This error message may also occur if the root directory is full, out of files, or if the filename is the same as a volume or directory, or a hidden (or system) file.

File is READ-ONLY
[EDLIN]

The file is designated read-only, so you may not change it.

File name must be specified
[EDLIN]

You did not specify a filename when you started EDLIN. You should type the EDLIN command followed by a filename.

File not found
[CHKDSK] [EDLIN] [FC] [FIND] [PRINT] [RECOVER]
[RENAME] [XCOPY]

MS–DOS could not find the file that you specified, or you tried to rename a file with a name already in the directory. Check to see that you entered the filename correctly.

File not in PRINT queue
[PRINT]

The file that you specified was not in the print queue, so you cannot remove it from the queue. Check to see that you entered the filename correctly.

Files cannot be added to this diskette
Unless the PACK (/P) switch is used
Set the switch (Y/N)?
[BACKUP]

The target disk does not have enough room for any of the files on the source disk without dividing them across disks. If you do not want to divide a file across disks, type **N** (for No). If your files are larger than will fit on one floppy disk, you must type **Y** (for Yes).

***** Files were backed up**
at *time* on *date* ***
[RESTORE]

This is an information message only.

FIND: Access denied
[FIND]

You cannot access the file. Make sure that the disk is not write–protected, read only, or locked.

FIND: File not found
[FIND]

MS–DOS could not find the file that you specified. Make sure you have typed the filename correctly.

FIND: Invalid number of parameters
[FIND]

You specified either too many or too few options in the command line.

FIND: Invalid Parameter
[FIND]

One of the switches you specified is wrong.

FIND: Read error in filename
[FIND]

The FIND command could not read the specified file.

FIND: Syntax error
[FIND]

Check to make sure that you have typed the command correctly.

First cluster number is invalid,
entry truncated
[CHKDSK]

The file directory entry contains an invalid pointer to the data area. If you specified the /F switch, the file is truncated to a zero-length file.

FIRST diskette bad or incompatible
[DISKCOMP]

DISKCOMP cannot recognize the format on the source disk. You should run CHKDSK to help you identify the problem.

Fixups needed - base segment hex:
[EXE2BIN]

The source (.EXE) file contained information indicating that a load segment is required for the file. You must specify the absolute segment address where the finished module is to be located.

Font File contents invalid
[MODE]

MS-DOS cannot use the contents of the font file specified. Make sure you are typing the name of the font file correctly. Retype the command. If this message is displayed again, your font file may have been altered or corrupted. Recopy this file from the master MS-DOS disk. Type the command again. This error may also cause existing selected code pages to be undefined. Use the mode command to prepare these code pages again, and to refresh them.

For cannot be nested
[MS–DOS]

You cannot nest FOR commands in a batch file.

Format another (Y/N)?
[FORMAT]

FORMAT displays this message when it has finished formatting a disk. Type **Y** (for Yes) if you want to format another disk, or type **N** (for No) if you don't. If you accidentally type **Y**, you can abort the format process by typing Ctrl–C in response to the message "Strike any key."

Format complete
[FORMAT]

FORMAT displays this message when it has finished formatting the disk in the specified drive.

Format failure
[FORMAT]

MS–DOS could not format the disk. This message is usually displayed with an explanation as to why the command failed.

Format not supported on drive *x*:
[FORMAT]

You cannot use FORMAT to format this drive. You may have specified device parameters that your computer cannot support.

Formatting while copying
[DISKCOPY]

DISKCOPY displays this message if the target disk has never been formatted.

General failure reading (or writing) drive *x*:
[MS–DOS]

This is a device error. See Appendix E, "Disk and Device Errors."

Graftabl needs DOS version 2.0 or later
[GRAFTABL]

You cannot use GRAFTABL with 1.xx versions of MS–DOS.

Graphics characters already loaded
[GRAFTABL]

The GRAFTABL command displays this message if you have already loaded the table of graphics characters into memory.

Graphics characters loaded
[GRAFTABL]

The GRAFTABL command displays this message after it loads the table of graphics characters into memory.

Hardware code pages:
Prepared code pages:
[MODE]

This message lists the current code pages prepared for the device specified.

Has invalid cluster, file truncated
[CHKDSK]

The file directory entry contains an invalid pointer to the data area. If you specified the /F switch, the file is truncated to a zero–length file.

Head: ***hhh*** **Cylinder:** ***ccc***
[FORMAT]

FORMAT displays the head and cylinder number of the track currently being formatted.

Illegal device name
[MODE]

Your computer does not recognize this device name.

Incompatible system size
[SYS]

The system files occupy more space on the source disk than is available on the target disk. You cannot use the SYS command to transfer the system files to this disk.

Incorrect APPEND Version
[MS–DOS]

You are not using the MS–DOS 3.3 APPEND command. You are using another incompatible version.

Incorrect DOS Version
[ATTRIB] [BACKUP] [CHKDSK] [DISKCOMP] [DISKCOPY] [EDLIN] [FC] [FIND] [FORMAT] [GRAPHICS] [JOIN] [KEYBXX] [LABEL] [MODE] [MORE] [PRINT] [RECOVER] [REPLACE] [RESTORE] [SHARE] [SORT] [SUBST] [SYS] [TREE] [XCOPY]

Some MS–DOS utilities will not run on older versions of the operating system, and many are written to run only on the exact version of MS–DOS that they were created for. You must use the correct version of MS–DOS to run this command.

Incorrect DOS Version, use DOS 2.00 or later
[LINK]

Some MS–DOS utilities will run only on MS–DOS version 2.00 or later versions.

Incorrect number of parameters
[JOIN] [SUBST]

You specified either too many or too few options in the command line.

Incorrect parameter
[ASSIGN] [SHARE]

One of the options you specified is wrong.

Infinite retry on parallel printer timeout
[MODE]

Your printer is probably off–line or not ready. If the printer appears to be ready, you may have to press the **Ctrl–Alt–Del** keys to reset the computer.

Insert backup diskette *n* into drive *x*:
[BACKUP] [RESTORE]

This message prompts you for the *n*th backup disk. Put the next disk into the specified drive. Be sure to label each backup disk in the appropriate order for use when restoring the files.

Insert destination disk in drive *x*:
and strike any key when ready
[SYS]

This message appears when you are uth a single disk drive. You shsk in theater or number key to begin processing.

Insert diskette for drive *x*:
and strike any key when ready
[MS-DOS]

This message appears when MS-DOS is copying and formatting. You should insert a disk in the appropriate drive and press any character or number key to begin processing.

Insert diskette with batch file
and press any key when ready
[MS-DOS]

The disk containing your batch file is not in the drive you originally specified. Reinsert the disk that contains the batch file in the appropriate drive.

Insert DOS diskette in drive *x*:
and strike ENTER when ready
[FORMAT]

You typed the FORMAT /S command, but the disk in the default drive does not contain MS-DOS system files. Insert a disk with the files IO.SYS and MSDOS.SYS in the drive specified and press any key.

Insert FIRST diskette into drive *x*:
[DISKCOMP]

This message prompts you for the first disk that you want to compare.

Insert last backup diskette in drive *x*:
Strike any key when ready
[BACKUP]

This message prompts you for the final backup disk. After you have put the final backup disk into the drive specified, press any alphanumeric key to continue the backup process.

Insert restore target diskette into drive *x*:
[RESTORE]

Restore displays this prompt if you are restoring files to a floppy. Put the target disk into the specified drive.

Insert SECOND diskette into drive *x*:
[DISKCOMP]

This message prompts you for the disk that you want to compare with the first disk.

Insert source disk
[BACKUP]

This message prompts you to put the source disk into the drive.

Insert SOURCE diskette into drive *x*:
[DISKCOPY]

This message prompts you to put the disk to be copied into the specified drive.

Insert system diskette in drive *x*:
and strike any key when ready
[SYS]

SYS needs a disk from which to read the IO.SYS and MSDOS.SYS files. Insert a system disk into the specified drive and press any character or number key to start the system copy process.

Insert TARGET diskette into drive *x*:
[DISKCOPY]

DISKCOPY displays this message to prompt you to place the target disk into the specified drive. If your computer has one floppy drive, this message prompts you to put the proper disk into the drive.

Insufficient disk space
[MS–DOS] [REPLACE] [SORT] [XCOPY]

The disk is full and does not contain enough room to perform the specified operation.

Insufficient memory
[BACKUP] [CHKDSK] [DISKCOMP] [DISKCOPY] [EDLIN] [REPLACE] [RESTORE] [SORT] [XCOPY]

There is not enough memory in your computer to perform the specified operation. Before retrying this operation, you must free memory by deleting files. In EDLIN, you may be able to free memory by typing a W (WRITE) command followed by an A (APPEND) command

Insufficient memory for system transfer
[FORMAT]

Your memory configuration is insufficient to transfer the MS-DOS system files IO.SYS and MSDOS.SYS with the FORMAT /S switch.

Insufficient room in root directory.
Erase files in root and repeat CHKDSK
[CHKDSK]

CHKDSK always recovers lost files into the root directory. In this case, your root directory is full. Delete some files in your root directory, or move them to another directory to make room for the lost files.

Intermediate file error during pipe
[MS-DOS]

The pipe operation uses temporary files on the disk that are deleted automatically once the piping process is complete. An error has occurred in one of these files. Make sure that there is enough room on the disk for the temporary file and that the disk is not write protected, and try the command again.

Internal error
[FC] [MODE] [SHARE]

This message indicates an error in the utility.

Invalid argument
[BACKUP] [FC] [RESTORE]

You have specified an invalid argument. Refer to Section 4, "MS-DOS Commands," for the correct syntax of the command, and try again.

Invalid baud rate specified
[MODE]

You have specified an incorrect baud rate. Valid choices are 110, 150, 300, 600, 1200, 2400, 4800, and 9600. You must specify at least the first two digits of the baud rate.

Invalid characters in volume label
[FORMAT] [LABEL]

The volume label should only contain up to 11 alphanumeric characters.

Invalid code page specified
[CHCP]

You selected an invalid code page number. Retype the command with the correct code page.

Invalid COMMAND.COM
Insert COMMAND.COM disk in default drive
and strike any key when ready
[MS–DOS]

The program you have just run used up almost all of memory. MS–DOS must now reload the COMMAND.COM file from disk. However, either MS–DOS cannot find COMMAND.COM on the disk, or the copy found is the incorrect version. Insert a disk that contains a copy of COMMAND.COM into the default drive (it must be the same version with which you started MS–DOS).

Invalid country code
[MS–DOS]

In your CONFIG.SYS file you have specified a country number that is not in the table of files configured in this version of MS–DOS. Country codes must be in the range 1–99.

Invalid country code or code page
[MS–DOS]

MS–DOS found an invalid country code or code page number in your CONFIG.SYS file. Correct the COUNTRY command line in your CONFIG.SYS file.

Invalid current directory
[CHKDSK]

Your disk has an invalid directory on it. You may be able to recover some of the files on this disk by copying them with the COPY command. Otherwise, you must replace the disk.

Invalid date
[DATE] [XCOPY]

You specified an invalid date in response to the date prompt. Enter a valid date. Refer to Section 4, "MS–DOS Commands," for the proper syntax of the date command.

Invalid Date/Time
[BACKUP]

You specified an invalid date with one of the BACKUP command switches. Refer to Section 4, "MS–DOS Commands," for the proper syntax of the BACKUP command.

Invalid device
[MS–DOS]

The device specified was not AUX, CON, NUL, or PRN.

Invalid device parameters from device driver
[FORMAT]

FORMAT displays this message when the number of hidden sectors is not evenly divisible by the number of sectors per track (that is, the partition does not start on a track boundary). This might happen if you tried to format a hard disk that previously had been formatted with MS–DOS 2.x without first running FDISK, or if you have set the device driver parameters incorrectly. Check the CONFIG.SYS file for incorrect DEVICE or DRIVPARM commands.

Invalid directory
[MS–DOS]

The directory you specified either does not exist or is invalid. Check to see that you entered the directory name correctly.

Invalid disk change reading (or writing) drive *x*:
[MS–DOS]

This is a device error. See Appendix E, "Disk and Device Errors."

Invalid drive in search path
[MS–DOS]

The drive does not exist.

Invalid drive or filename
[EDLIN] [RECOVER]

You did not type a valid drive name or filename. Enter a valid drive name or filename.

Invalid drive specification
[BACKUP] [CHKDSK] [DISKCOMP] [DISKCOPY] [FORMAT] [LABEL] [PRINT] [REPLACE] [RESTORE] [SYS] [TREE] [XCOPY]

The drive is incorrect or does not exist. Enter a valid drive name.

Invalid environment size specified
[COMMAND]

You gave an invalid number of bytes with the /E switch. You must specify a number between 128 and 32768 (bytes).

Invalid keyboard code specified
[KEYB]

You selected an invalid keyboard code with the KEYB command. Retype the command with the correct keyboard code.

Invalid language specified
[KEYB]

You typed an invalid keyboard code with the KEYB command. See the KEYB command in Section 4, "MS–DOS Commands", for a list of valid keyboard codes. Retype the command using a valid keyboard code.

Invalid number of parameters
[ATTRIB] [BACKUP] [FC] [FIND] [RECOVER] [RESTORE] [XCOPY]

Either you did not specify an option or string, or you specified the wrong number of options in the command line.

Invalid parameter(s)
[BACKUP][CHKDSK][DISKCOMP][DISKCOPY][EDLIN][FIND]
[FORMAT][JOIN][MODE][PRINT][REPLACE][RESTORE][SORT]
[SUBST][SYS][TREE][XCOPY]

One of the switches you specified is wrong or does not exist. Refer to Section 4, "MS-DOS Commands," to make sure you are using the correct switches.

Invalid path, not directory, or directory not empty
[MS-DOS]

You are unable to remove the directory requested for one of the specified reasons.

Invalid path (or file not found)
[ATTRIB][BACKUP][COPY][RESTORE][TREE][XCOPY]

You have entered a pathname or filename that does not exist. Enter a valid pathname or filename with the command.

Invalid path or parameter
[APPEND]

You specified a file or directory that does not exist. Enter a valid pathname or filename with the APPEND command.

Invalid signature in COUNTRY.SYS file
[SELECT]

SELECT reads the COUNTRY.SYS file to verify the country code. SELECT quits if it cannot find a proper file header or a specific country code.

Invalid signature in KEYBOARD.SYS file
[SELECT]

SELECT reads the KEYBOARD.SYS file to verify the keyboard code. SELECT quits if it cannot find the proper file header or a specific keyboard code.

Invalid STACK parameter
[MS-DOS]

The syntax of the STACK command in your CONFIG.SYS file includes an invalid parameter. See Appendix C, "How to Configure Your System", for the correct syntax of that configuration command.

Invalid sub-directory entry
[CHKDSK]

The subdirectory that you specified either does not exist or is invalid. Check to see that you typed the subdirectory name correctly.

Invalid syntax
[MS-DOS]

You used the wrong syntax when typing a command. See Chapter 4, "MS-DOS Commands", for the correct syntax.

Invalid syntax on DISPLAY.SYS code page driver
[MS-DOS]

You used the wrong syntax when you typed the DEVICE command in your CONFIG.SYS file to load DISPLAY.SYS. See Appendix C, "How to Configure Your System", for the correct syntax of DEVICE. Also see Appendix D, "Installable Device Drivers", for information about the DISPLAY.SYS installable device driver.

Invalid syntax on PRINTER.SYS code page driver
[MS-DOS]

You used the wrong syntax when you typed the DEVICE command in your CONFIG.SYS file to load PRINTER.SYS. See Appendix C, "How to Configure Your System", for the correct syntax of DEVICE. Also see Appendix D, "Installable Device Drivers", for information about the PRINTER.SYS installable device driver.

Invalid syntax on PRINTER.SYS code page switching device drivers
[MODE]

You used the wrong syntax when you typed the DEVICE command in your CONFIG.SYS file to load PRINTER.SYS. See Appendix C, "How to Configure Your System", for the correct syntax of DEVICE. Also see Appendix D, "Installable Device Drivers", for information about the PRINTER.SYS installable device driver.

Invalid time
[TIME]

You specified an invalid time. Refer to Section 4, "MS-DOS Commands," for the correct syntax, and try the command again.

Invalid Volume ID
[FORMAT]

FORMAT displays this message if you enter a volume label that doesn't match the label on the hard disk you want to format. It then quits the format process. Use the VOL command to find out what the volume label for the hard disk is, then try the command again.

Invalid working directory
Process cannot continue
[CHKDSK]

The current directory of the disk being checked is damaged and unusable.

KEYB has not been installed
[MS-DOS]

No alternate keyboard code has been installed for your system. If you want to use keyboard code other than the default U.S. (QWERTY) keyboard, use the KEYB command to install it.

Label not found
[MS-DOS]

Your batch file contains a GOTO command to a nonexistent label.

Last backup diskette not inserted
Insert last backup diskette in drive *x*:
Strike any key when ready
[BACKUP]

This message prompts you for the final backup disk. After you have put the final backup disk into the drive specified, press any alphanumeric key to continue the backup process.

**** Last file not backed up ****
[BACKUP]

BACKUP could not back up the last file on the disk. This message may occur if there is no more room on the target disk. It may also occur if there was an error in the source file, or on the target disk. You may have to back up this file separately to another disk.

Line too long
[EDLIN]

During an EDLIN R (REPLACE) command, the string given as the replacement caused the line to expand beyond the limit of 253 characters. You should divide the long line into two lines and retry the R command.

List output is not assigned to a device
[PRINT]

When you first type the PRINT command, MS–DOS asks you what device you want to specify as a printer. This message appears if PRINT is set up for a device that does not exist.

Lock violation reading (or writing) drive ***x*****:**
[MS–DOS]

This is a device error. See Appendix F, "Disk and Device Errors."

Logging to file x
[BACKUP]

The BACKUP command is writing a backup log to the file specified.

Logical DOS drive created, drive letters changed or added
[FDISK]

You have created or revised one or more logical drives.

x **lost cluster(s) found in** *y* **chains**
Convert lost chains to files (Y/N)?
[CHKDSK]

CHKDSK displays this message if it finds information on the disk that isn't allocated properly in the disk's File Allocation Table. If you type Y in response to this prompt, CHKDSK recovers the lost blocks it found when checking the disk. CHKDSK then creates a proper directory entry and a file for each lost chain with the filename of the form: filennnn.chk. If you did not specify the /F switch, CHKDSK displays: "*x* bytes would be freed." If you type N, CHKDSK frees the lost blocks so that they can be reallocated and does not recover any data that was in those lost blocks. If you did not specify the /F switch, CHKDSK does nothing.

LPT#: not redirected
[MODE]

MODE could not redirect the parallel printer port. Check to see whether you have specified the proper options.

LPT#: redirected to COM#:
[MODE]

Output on the parallel printer port will now be sent to this asynchronous communications port.

LPT#: set for 80
[MODE]

The parallel printer port has been set for 80 columns.

LPT#: set for 132
[MODE]

The parallel printer port has been set for 132 columns.

Maximum available space for partitions is xxx cylinders
[FDISK]

This is an informational message.

Maximum number of logical DOS drives installed
[FDISK]

You have installed the maximum number of logical DOS drives allowed by MS-DOS. You may not create any more logical DOS drives.

Memory allocation error.
Cannot load MS-DOS, system halted
[MS-DOS]

Restart MS-DOS. If this error persists, make a new copy of the MS-DOS disk from your backup copy of the system disk.

Missing from the file is either the device ID or the code page
[MODE]

The code page specified is not supported in the Code Page Information (.CPI) file, or the .CPI file does not support the printer specified. For a list of valid CP list values in the MODE command, see Section 4, "MS-DOS Commands."

MODE fff code page function complete
[MODE]

This message is informational only.

--More–
[MORE]

Press the space bar to view more of the file or directory.

MORE: Incorrect DOS version
[MORE]

The MORE command does not run on MS–DOS versions before 2.0.

Must specify destination line number
[EDLIN]

You did not specify the destination line number for an EDLIN C (COPY) or M (MOVE) command. Retype the command with a destination line number.

Must specify ON or OFF
[MS–DOS]

The command requires either an ON or an OFF argument.

Name of list device [PRN]:
[PRINT]

This prompt appears the first time that PRINT is run and the /D switch is not specified. You can specify the name of any valid device, which then becomes the print output device. If you press the **Enter** key, MS–DOS uses the default list device PRN.

New file
[EDLIN]

EDLIN prints this message if it does not find a file with the name you specified. If you are creating a new file, ignore this message. If you do not intend to create a new file, check to see whether you have correctly typed the name of the file that you wish to edit.

NLSFUNC already installed
[NLSFUNC]

NLSFUNC stays resident in memory once it is initialized. You have already loaded it into memory.

No APPEND
[APPEND]

No paths have been appended. If you would like to append a path for data files, use the APPEND command.

No appended directories
[APPEND]

You did not specify a path with the APPEND command.

No code page has been selected
[CHCP]

No code pages have been selected for the system. If you would like to select a code page, use the CHCP command.

No COM: ports
[MODE]

Your computer does not have an asynchronous communications (serial) port.

No files added (or replaced)
[REPLACE]

The REPLACE command did not add or replace any files.

No files found filename
[REPLACE]

REPLACE could not find matching source or target files.

No free file handles.
Cannot start COMMAND.COM, exiting
[MS-DOS]

Restart MS-DOS. If this message persists, increase the files command value in the CONFIG.SYS file.

No logical drives defined
[FDISK]

There are no logical drives defined for your system.

No paper error writing device *dev*
[MS–DOS]

This is a device error. See Appendix F, "Disk and Device Errors."

No path
[PATH]

You typed PATH and pressed the **Enter** key to find out what your search path is, but you didn't set a command search path.

No primary DOS partition to delete
[FDISK]

You have selected the FDISK option to delete your primary DOS partition, but that partition does not exist.

No room for system on destination disk
[SYS]

There is not enough room for the system files on the target disk. Delete some files to make room for the system files or use another disk. You may need to reformat the disk to put the system on it.

No room in directory for file
[EDLIN]

You tried to create or save a file to the root directory, but it is either full, or you specified an invalid disk drive or filename. Check the command line that you used to start EDLIN for an invalid filename or disk drive entry. If your command contains no invalid entries, you should run the CHKDSK program for the specified disk drive. If the status report shows that the disk directory is full, and if there is still enough memory left on the disk, you may be able to create the file in a subdirectory. (This is because subdirectories are not limited in size as is the root directory.) Otherwise, remove the disk and replace it with another formatted disk.

No room in root directory
[LABEL]

There is not enough room in the root directory for a volume label. Delete or move some of the files from the root directory to make room for the volume label.

No source drive specified
[BACKUP]

You must specify a source drive.

No space left on device
[BACKUP] [FC] [RESTORE]

You cannot back up or restore any more files, and you cannot send any more output from a file comparison to your disk because the target disk is now full. You should probably delete some of the files on the disk to make more room.

No space to create logical drive
[FDISK]

You are trying to create a logical drive, but there is no space available to do so.

No sub-directories exist
[TREE]

You have specified the /S switch, but the directory does not contain sub-directories.

No such file or directory
[BACKUP] [FC] [RESTORE]

One or more of the files or directories that you specified does not exist.

Non-DOS disk error reading (or writing) drive *x*:
[MS-DOS]

This is a device error. See Appendix F, "Disk and Device Errors."

No version of Graphic Character Set Table is loaded
[GRAFTABL]

For information only.

Non-standard version of Graphic Character Set Table is already loaded
[GRAFTABL]

MS-DOS cannot recognize the current table of graphics characters because it has been modified since it was loaded.

Non-system disk or disk error
Replace and strike any key when ready
[FORMAT] [SYS]

Replace the disk with the proper disk and press any alphanumeric key to continue.

***** Not able to back up (or restore) file *****
[BACKUP]

This message may occur if there was an error in the source file or on the target disk. Use the CHKDSK command on the source disk to see if you can determine the problem.

Not a graphics printer file
[GRAPHICS]

The file you are printing does not contain graphics.

Not enough memory
[JOIN] [SHARE] [SUBST]

There is not enough memory for MS-DOS to run the command.

Not enough room to merge the entire file
[EDLIN]

There was not enough room in memory to hold the file during an EDLIN T (TRANSFER) command. You must free some memory by writing some files to a disk or by deleting some files before transferring this file.

Not found
[EDLIN]

You specified an EDLIN S (SEARCH) or R (REPLACE) command that was unable to find a further occurrence of the specified search or replace string.

Not ready error reading (or writing) drive *x*:
[MS-DOS]

This is a device error. See Appendix F, "Disk and Device Errors."

O.K.?
[EDLIN]

This prompt occurs during EDLIN S (SEARCH) or R (REPLACE) command processing. If you press any key except **Y** (for Yes) or the **Enter** key, the search or replace process continues.

One or more CON code pages invalid for given language
[KEYB]

KEYB examined all prepared code pages, and has found that at least one code page is incompatible for your screen console device (CON). This is only a warning to let you know that your keyboard and screen console device are working from different code pages.

Only non-bootable partitions exist
[FDISK]

None of the partitions left can boot MS-DOS.

Only partitions on drive 1 can be made active
[FDISK]

You are trying to create an active partition on a hard disk other than that found on the first hard disk drive. This is not allowed.

Out of environment space
[COMMAND] [MS-DOS]

There is not enough room in the program environment to accept more data. To increase the size of the existing environment, use the /E switch with the COMMAND command or remove some of the existing environment variables by using the set command.

Parameters not compatible
[FORMAT] [REPLACE]

You have specified switches that cannot be used together.

Parameters not compatible with fixed disk
[FORMAT]

You have used a switch that is not compatible with the specified drive.

Parameters not supported
[MS–DOS] [FORMAT]

You have specified parameters that MS–DOS does not support.

Parameters not supported by Drive
[FORMAT]

FORMAT displays this message when the device driver for this drive does not support Generic IOCTL function requests.

Partition selected (x) is not bootable, active partition not changed
[FDISK]

You are trying to change active partitions, but MS–DOS cannot be booted from the partition selected.

Path(name) too long
[PRINT] [REPLACE] [XCOPY]

The pathname you specified was too long. You may have to change directories to use this command with files in deep subdirectories.

Path not found
[CHKDSK] [REPLACE] [SUBST] [XCOPY]

You specified an invalid pathname.

Press any key to begin adding (replacing) file(s)
[REPLACE]

When you specify the /W switch, REPLACE displays this message to prompt you to start replacing files.

Press any key to begin formatting ***x*:**
[FORMAT]

This prompt is issued before you format a disk. Press any key to begin the format process. Or, if you wish to end this command, press **Ctrl–C**.

Press any key to begin recovery of the
n **file(s) on drive** ***x*****:**
[RECOVER]

This prompt is issued before you recover a disk or file. Press any key to begin the recovery. Recovered files are named file*nnnn*.REC. If you wish to end this command, press **Ctrl-C**.

Press any key when ready . . .
[DISKCOMP] [DISKCOPY]

This prompt gives you time to insert the appropriate disks before copying them. When you have inserted the disks into the appropriate drives, press any key to begin the diskcopy process. Or, if you wish to end this command, press **Ctrl-C**.

Previously prepared code page replaced
[MODE]

This command changed the selected code page for a specific device by using another prepared code page.

Primary DOS partition already exists
[FDISK]

You are trying to create a primary DOS partition, but one already exists. If there is space available on your hard disk, try to create an extended DOS partition instead.

Primary DOS partition created
[FDISK]

You have successfully created a primary DOS partition on your disk.

Primary DOS partition deleted
[FDISK]

You have deleted the primary DOS partition from your disk.

Printer error
[MODE]

The printer is off, or is not ready.

Printer lines per inch set
[MODE]

MODE has set the number of lines per inch for the printer.

PRINT queue is empty
[PRINT]

There are no files writing to be printed.

PRINT queue is full
[PRINT]

There is only room for 10 files in the list of files waiting to be printed. You can make room for more by using the PRINT /Q switch. The limit is 32 files.

Probable non-DOS disk
Continue (Y/N)?
[CHKDSK]

The disk you are using is not recognized by this version of MS-DOS. The disk was either created by another system with a format that is not supported on this version of MS-DOS, or it is not an MS-DOS disk. Do not continue processing if CHKDSK returns this message for a floppy disk. If this message returns for a hard disk, the information describing the characteris tics of the disk to MS-DOS has been destroyed. In this case, you may continue CHKDSK processing by typing Y (for Yes). This error may mean that the File Allocation Table (FAT) is bad and that the disk is unusable.

Processing cannot continue
[CHKDSK]

There is not enough memory in your machine to run CHKDSK for this disk. You must obtain more memory to run CHKDSK.

Program too big to fit in memory
[MS-DOS]

You need more memory to run your application. It is possible that some programs you have run are still using some memory. You may try to restart MS-DOS; however, if you still receive this message, you still need more memory.

Read error, COUNTRY.SYS
[MS-DOS]

MS-DOS cannot read the COUNTRY.SYS file. Retry the command. If you get the same message, the COUNTRY.SYS file is probably corrupted. Restore the file from backup.

Read error in filename
[EDLIN] [FIND]

MS-DOS could not read the entire file.

Read fault error reading drive *x*:
[MS-DOS]

This is a device error. See Appendix F, "Disk and Device Errors."

Read error, KEYBOARD.SYS
[MS-DOS]

MS-DOS cannot read the KEYBOARD.SYS file. Retry the command. If you get the same message, the KEYBOARD.SYS file is probably corrupted. Restore the file from backup.

Reading source file(s)...
[XCOPY]

XCOPY is now reading the source files that you specified.

Reinsert diskette for drive *x*:
[FORMAT]

Reinsert the disk being formatted in the indicated drive.

Replace filename? (Y/N)
[REPLACE]

REPLACE displays this prompt if you specify the /W switch. Press **Y** if you want to replace the existing file, or **N** if you do not want to replace the file.

Replace the file (Y/N)?
[RESTORE]

The file that you want to restore from backup already exists on your target disk. Type **Y** (for Yes) and press the **ENTER** key to overwrite the file. Type **N** (for No) and press the **ENTER** key if you do not want to replace the file on your target disk with the file from the backup disk.

Replacing filename
[REPLACE]

REPLACE displays this prompt to let you know that it is replacing this file that exists on your disk.

Requested logical drive size exceeds the maximum available space
[FDISK]

You are trying to create a logical drive that is larger than the space available.

Requested partition size exceeds the maximum available space
[FDISK]

You are trying to create a partition on your hard drive that is larger than the space available.

Requested Screen Shift out of range
[MODE]

You cannot shift the display any farther.

Resident part of PRINT installed
[PRINT]

This is the first message that MS–DOS displays when you issue the PRINT command. It means that to process the PRINT command with other processes, available memory has been reduced by several thousand bytes.

Resident portion of MODE loaded
[MODE]

Part of the MODE program is now resident in memory, and available memory has been reduced by several thousand bytes.

Resident portion of NLSFUNC loaded
[NLSFUNC]

NLSFUNC stays resident in memory once it is initialized. This informational message lets you know that you have already loaded NLSFUNC.

Restore file sequence error
[RESTORE]

You have restored files in the wrong order. You must insert the backup disks in the same order that they were backed up.

***** Restoring files from drive *x*: *****
Diskette: *n*
[RESTORE]

This message is displayed during the restore process.

Resynch failed. Files are too different
[FC]

FC compares what can be loaded into memory. If no lines match in the portion of the files in the buffer space, FC displays this message.

Same drive specified more than once
[FASTOPEN]

You tried to activate FASTOPEN for the same drive more than once. There is no need to reactivate it for the same drive.

SECOND diskette bad or incompatible
[DISKCOMP]

The second disk does not contain the same format as the first disk, or DISKCOMP does not recognize the format of the second disk. You should run CHKDSK to help you identify the problem.

Sector not found error reading (or writing) drive *x*:
[MS–DOS]

This is a device error. See Appendix F, "Disk and Device Errors."

Sector size too large in file filename
[MS–DOS]

The specified device driver loaded by CONFIG.SYS uses a sector size larger than that of any other device driver on the system. You cannot run this device driver.

Seek error reading (or writing) drive *x*:
[MS–DOS]

This is a device error. See Appendix F, "Disk and Device Errors."

SHARE already installed
[SHARE]

SHARE can only be installed once.

Sharing violation reading drive *x*:
[MS–DOS]

This is a device error. See Appendix E, "Disk and Device Errors."

SORT: Incorrect DOS version
[SORT]

SORT does not run on MS–DOS versions before 2.0.

SORT: Insufficient disk space
[SORT]

The disk is full.

SORT: Insufficient memory
[SORT]

There is not enough memory to run the SORT program.

Source and target drives are the same
[BACKUP][RESTORE]

You specified the same drive for the source and target disks.

Source disk is Non–removable
[BACKUP]

This is an informational message indicating that the source disk is a hard disk.

Source does not contain backup files
[RESTORE]

You are attempting to restore files from a disk that does not contain backup files.

Source is a floppy (or hard) disk
[RESTORE]

This is an informational message only.

Source path required
[REPLACE]

You did not specify a source path for the REPLACE command.

Specified drive does not exist, or is non-removable
[DISKCOMP] [DISKCOPY]

You cannot compare or copy hard disks with this command. You must specify the name of a valid floppy drive.

Specified MS-DOS search directory bad
[MS-DOS]

The SHELL command in the CONFIG.SYS file is incorrect. Make sure that the COMMAND.COM file exists and that MS-DOS can find it.

Strike a key when ready...
[MS-DOS]

This prompt occurs during command processing and is always accompanied by another message. This message is also displayed if you have inserted a PAUSE command in a batch file. Usually, MS-DOS asks you to insert disks into appropriate drives before this prompt. To begin command processing, press any character, any number key, the spacebar, or the Enter key.

Syntax error
[ATTRIB] [FIND] [MS-DOS]

You have entered a command incorrectly. Check to make sure you have typed the command correctly. Remember to enclose the FIND command string in double quotation marks.

System transferred
[FORMAT] [SYS]

The system files were transferred during FORMAT or SYS command processing.

Target cannot be used for backup
[BACKUP]

Either the target disk has an unrecognizable format, or it is bad. Do not use the disk, or try to format the disk with the FORMAT command, or run CHKDSK on it to determine the problem.

Target disk is Non-removable
[BACKUP]

This is an informational message that the target disk is a hard disk.

Target diskette is write protected
[DISKCOPY]

The target disk either has a write-protect tab on it, or it does not have a write-protect notch. If you want to destroy any existing information on the disk, remove the write-protect tab and give the command again. If the disk does not have a write-protect notch, you cannot use it as a target disk.

Target diskette may be unusable
[DISKCOPY]

Either the target disk has an unrecognizable format, or it is bad. Try to format the disk with the FORMAT command, or run CHKDSK on it to determine the problem.

Target is a floppy (or hard) disk
[BACKUP]

This is an informational message only.

Target is full
[RESTORE]

There is no more room on the target disk for restored files. You must delete some of the files on the disk to make room for these files, or use another disk.

Target is Non-Removable
[RESTORE]

This is an informational message only.

Terminate batch job (Y/N)?
[MS-DOS]

If you press Ctrl-C while in batch mode, MS-DOS asks you whether or not you wish to end batch processing. Press **Y** (for Yes) to end processing, or **N** (for No) to continue.

The current active keyboard table is xx with code page: yyy The current active CON code page is zzz
[KEYB]

This is an informational message that shows the current keyboard code, code page for the system and code page for the console (screen).

The last file was not restored
[RESTORE]

There was not enough room on the target disk for the file, or the last file was bad. Use the CHKDSK command to determine the problem.

The only bootable partition on drive 1 is already marked active
[FDISK]

You are trying to change the active partition. The active partition must reside on the first hard disk drive on your system and must be bootable. The only bootable partition on the first hard disk drive is already the active partition.

Too many drive entries
[FASTOPEN]

You can use FASTOPEN with up to four hard drives. You have tried to specify a fifth drive.

Too many files open
[EDLIN] [LABEL]

MS-DOS could not open the .BAK file or write the volume label due to the lack of available system file handles. Increase the value of the files command in the CONFIG.SYS file.

Too many name entries
[FASTOPEN]

The total number of entries specified for a drive exceeded the maximum of 999.

Too many open files
[BACKUP] [FC] [RESTORE] [XCOPY]

MS-DOS could not open the files that you want to compare due to the lack of available system file handles. Increase the value of the files command in the CONFIG.SYS file.

Track 0 bad - disk unusable
[FORMAT]

The FORMAT command can accommodate defective sectors on the disk, except for those near the beginning. You must use another disk.

Unable to create directory
[MKDIR] [XCOPY]

MS–DOS could not create the directory you specified. Check to see that there is not a name conflict. You may have a file with the same name, or the disk may be full.

Unable to create KEYB table in resident memory
[KEYB]

MS–DOS tried to create a country–specific table for the keyboard code specified, but failed. Check the amount of available memory. There may not be enough memory available to create this table.

Unable to erase
[BACKUP]

BACKUP could not erase the files on the target disk. Check to see that the files on the backup disk are not read–only, and that the disk is not write–protected.

Unable to shift Screen
[MODE]

MODE is unable to shift the test pattern on the screen any farther.

Unexpected DOS Error *n*
[REPLACE]

An unexpected error *n* occurred, where *n* is the MS–DOS error number.

Unrecognized command in CONFIG.SYS
[MS–DOS]

There is an invalid command in your CONFIG.SYS file. Refer to Appendix C, "How to Configure Your System," for a list of valid statements.

Unrecognized printer
[GRAPHICS]

You are using an invalid printer. Check to see whether you entered the command properly, or refer to Section 4, "MS–DOS Commands," to make sure that you have specified a valid printer name.

Unrecognized printer port
[GRAPHICS]

The printer device name that you specified was invalid. You may need to set the printer port by using the MODE command.

Unrecoverable error in directory
Convert directory to file (Y/N)?
[CHKDSK]

This message is displayed if CHKDSK is unable to correct an error in a directory. If you respond **Y** (for Yes) to this prompt, CHKDSK converts the bad directory into a file. You can then fix the directory or delete it. If you respond **N** (for No) to this prompt, you may not be able to write to or read from the bad directory.

Unrecoverable read (or write) error on drive *x*:
[MS-DOS]

This is a device error. See Appendix E, "Disk and Device Errors."

usage: fc [/a] [/b] [/c] [/l] [/lb n] [/w]
[t] [/n] [/NNNN] file1 file2
[FC]

One of the switches that you have specified is invalid.

VERIFY is off (or on)
[MS-DOS]

This message tells you the current setting of the VERIFY command.

Volume in drive *x*: has no label
[DIR][LABEL][VOL]

This is an informational message displayed in response to the DIR, LABEL, or VOL command.

Volume in drive *x*: is filename
[DIR][LABEL][VOL]

This is an informational message displayed in response to the DIR, LABEL, or VOL command.

Volume label (11 characters, ENTER for none)?
[FORMAT] [LABEL]

This message is displayed when you specify the LABEL command, or the /V switch in the FORMAT command. Type a volume label, or press the **Enter** key to indicate that you do not want a volume label for this disk. It's a good idea, though, to specify a volume label to help you identify your disks.

WARNING, ALL DATA ON NON-REMOVABLE DISK DRIVE *x*: WILL BE LOST!
Proceed with Format (Y/N)?
[FORMAT]

This message appears when you try to format a hard disk that already contains data. If you press Y (for Yes) the data on the disk will be erased. If you do not want the files on your hard disk erased, press **N** (for No). Copy the files to a floppy disk and repeat the FORMAT command.

Warning! Date in the extended DOS partition could be destroyed. Do you wish to continue.......? [n]
[FDISK]

You are trying to delete an extended DOS partition. Be sure this is what you want to do. If you do want to delete the extended DOS partition specified, type Y (for Yes) and press the **ENTER** key. If not, the default response N is already typed. Press the **ENTER** key.

Warning! Date in the primary DOS partition could be destroyed. Do you wish to continue.......? [n]
[FDISK]

You are trying to delete the primary DOS partition. If you are sure you want to delete your primary DOS partition, type Y and press the **ENTER** key. If not, press the **ENTER** key; MS–DOS will not delete the partition.

Warning – directory full
[RECOVER]

The root directory is too full for RECOVER processing. Delete some files in the root directory to free space for the recovered files, and try the command again.

Warning! Diskette is out of sequence
Replace diskette or continue if okay
Strike any key when ready
[RESTORE]

You should restore the diskettes in the order that you backed them up.

Warning! File filename is a hidden (or read-only) file
Replace the file (Y/N)?
[RESTORE]

This message prompts you as to whether you want to replace a hidden or read-only file. Type **Y** (for Yes) if you want to restore the hidden or read-only file from the backup disk. Type **N** (for No) if you do not want to restore this file.

Warning! File filename was changed after it was backed up
Replace the file (Y/N)?
[RESTORE]

This message prompts you as to whether you want to replace a backup file that has been changed. Type **Y** (for Yes) if you want to restore this file, or **N** (for No) if you do not.

Warning! Files in the target drive
\BACKUP (or root) directory will be erased
[BACKUP]

BACKUP found files in the target drive, and you did not specify the /A switch to append files.

Warning! No files were found to back up
[BACKUP]

BACKUP did not find any files to back up on the disk you specified.

Warning! No files were found to restore
[RESTORE]

RESTORE did not find the file that you wanted to restore from the backup disk.

Warning: Read error in EXE file
[EXE2BIN]

The amount read was less than the size of the header. This is a warning message only.

Warning! The partition marked active is not bootable
[FDISK]

The active DOS partition must be bootable.

Write fault error writing drive ***x*****:**
[MS–DOS]

This is a device error. See Appendix F, "Disk and Device Errors."

Write protect error writing drive ***x*****:**
[MS–DOS]

This is a device error. See Appendix F, "Disk and Device Errors."

Appendix H

Configuring Your Hard Disk (FDISK)

FDISK is a fixed (hard) disk utility that partitions a large hard disk.

Use FDISK to prepare a hard disk by performing one or more of the following tasks:

- create up to eight MS–DOS partitions
- create an extended MS–DOS partition
- specify the active partition
- delete a partition
- display partition data.

In addition, if you have more than one hard disk, you can select another disk drive to partition.

Notes

To run FDISK, you must have the following:

- A computer with one or more hard disk drives.
- MS–DOS version 3.2 or higher.

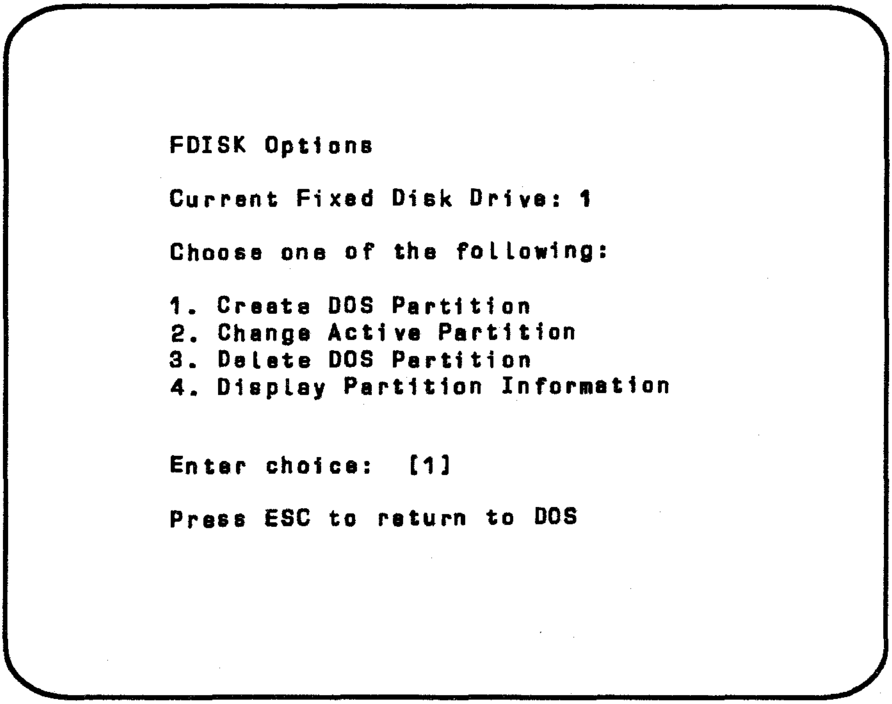
Syntax

FDISK

Comments

FDISK automatically assigns drive letters to each partition as follows: Partition 1 = C, Partition 2 = D, up to Partition 8 = J.

After you enter the FDISK command, the following option menu appears on your screen.



```
FDISK Options

Current Fixed Disk Drive: 1

Choose one of the following:

1. Create DOS Partition
2. Change Active Partition
3. Delete DOS Partition
4. Display Partition Information


Enter choice:  [1]

Press ESC to return to DOS
```

If your computer has more than one hard disk, FDISK lists a fifth option: Select Next Fixed Disk Drive.

SETTING THE PARTITION SIZE

When you use FDISK to create an MS–DOS partition, you are asked to specify the partition size in *cylinders*. The 40–MB hard disk contains 614 cylinders. The 66–MB hard disk contains 1000 cylinders.

1 cylinder	= 68 KB
100 cylinders	= 6.6 MB (approximately)

Previously, the maximum size of an MS–DOS partition was 32 MB. To use all of the storage capacity of the 40–MB hard disk, for example, the disk had to contain more than one partition. NEC has customized MS–DOS 3.3 to allow you to create partitions larger than 32 MB. Some possible configurations of a 40–MB hard disk are shown in Figure G–1.

Configuration 1

DRIVE LETTER	SIZE
C:	40 MB (614 cylinders)

Configuration 2

DRIVE LETTER	SIZE
C:	32 MB (481 cylinders)
D:	8 MB (133 cylinders)
	40 MB (614 cylinders)

Configuration 3

DRIVE LETTER	SIZE
C:	20 MB (307 cylinders)
D:	20 MB (307 cylinders)
	40 MB (614 cylinders)

Configuration 4

DRIVE LETTER	SIZE
C:	32 MB (481 cylinders)
D:	8 MB (133 cylinders)
C:	20 MB (307 cylinders)
D:	10 MB (154 cylinders)
E:	10 MB (153 cylinders)
	40 MB (614 cylinders)

Configuration 5

DRIVE LETTER	SIZE
C:	20 MB (307 cylinders)
D:	20 MB (307 cylinders)
C:	10 MB (154 cylinders)
D:	10 MB (154 cylinders)
E:	10 MB (153 cylinders)
F:	10 MB (153 cylinders)
	40 MB (614 cylinders)

Configuration 6

DRIVE LETTER	SIZE
C:	5 MB (77 cylinders)
D:	5 MB (77 cylinders)
E:	5 MB (77 cylinders)
F:	5 MB (77 cylinders)
G:	5 MB (77 cylinders)
H:	5 MB (77 cylinders)
I:	5 MB (76 cylinders)
J:	5 MB (76 cylinders)
	40 MB (614 cylinders)

Configuration 7

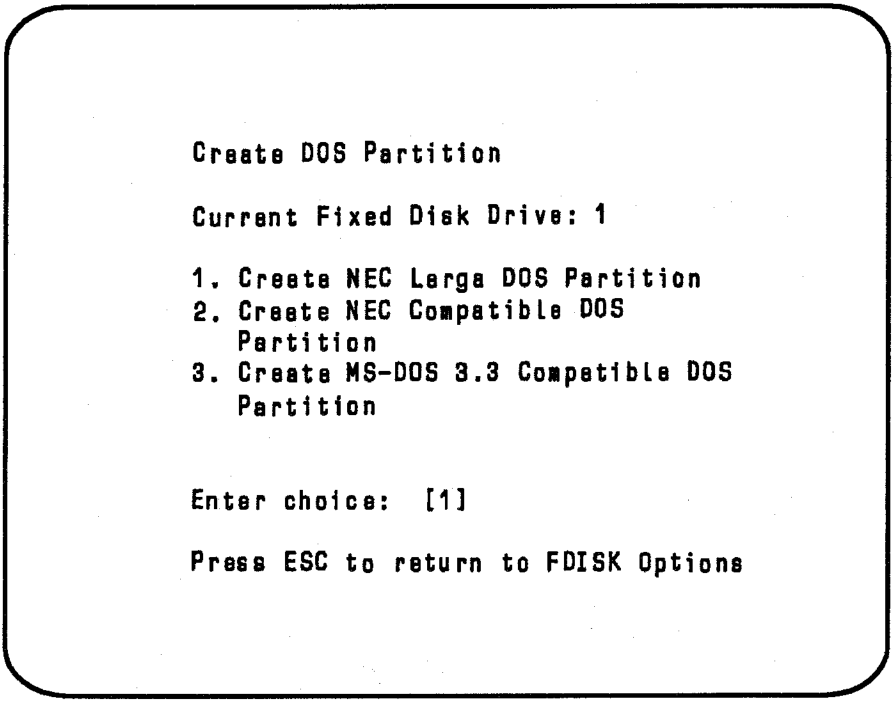
DRIVE LETTER	SIZE
C:	20 MB (307 cylinders)
Extended Partition	20 MB (307 cylinders)
D:	10 MB (154 cylinders)
E:	5 MB (77 cylinders)
F:	5 MB (77 cylinders)

Figure G–1 Sample Configurations of MS–DOS Partition Sizes

OPTION 1: CREATING A DOS PARTITION

Option 1 permits you to divide a hard disk into regions called partitions. Each partition can use a different operating system, if space allows.

To create a partition that uses MS–DOS as its operating system, enter **1** in the Options menu. A screen similar to the following example appears.



```
Create DOS Partition

Current Fixed Disk Drive: 1

1. Create NEC Large DOS Partition
2. Create NEC Compatible DOS
   Partition
3. Create MS-DOS 3.3 Compatible DOS
   Partition


Enter choice:  [1]

Press ESC to return to FDISK Options
```

MS–DOS 3.3 offers several new options. The first option is to create a partition greater than 32 MB (NEC Large DOS Partition). This NEC customization allows you to use the whole disk (40 MB) as one drive (C:). This new feature is available with NEC MS–DOS 3.3 only. (See Configuration 1.)

The second option is to create NEC compatible multiple partitions (Configurations 2 through 6). This allows you to create up to 8 bootable partitions, less than 32 MB each, that are compatible with NEC MS–DOS 3.1 and 3.2. You may also combine Options 1 and 2 for a maximum of 8 bootable partitions.

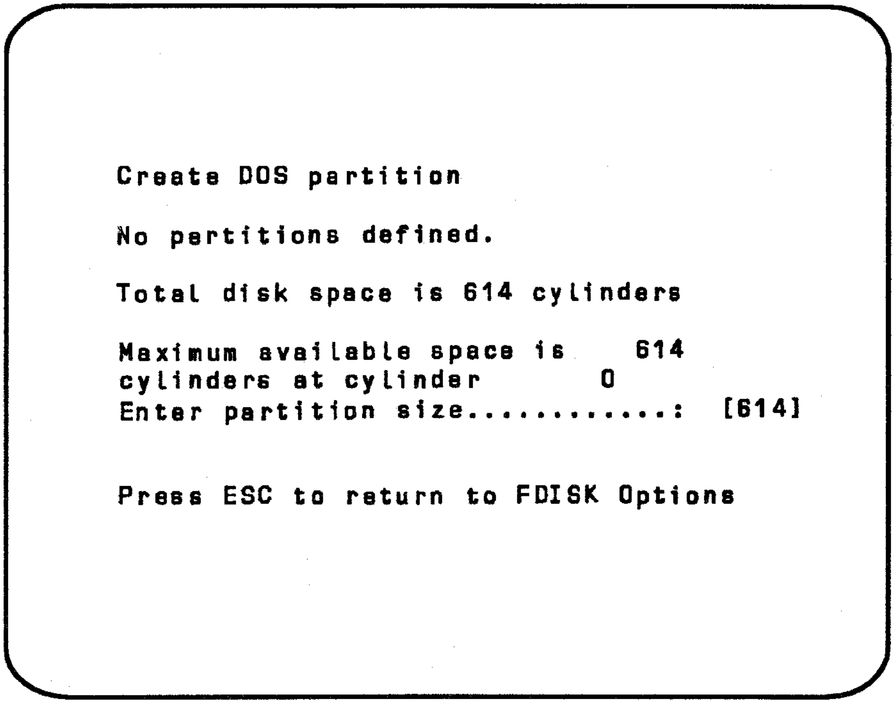
Example: 130 MB

Drive	Size
C:	30 cylinders
D:	50 cylinders
E:	50 cylinders

The third option is to create a standard environment compatible with MS–DOS 3.3 (Configuration 7). This allows you to create 1 bootable partition no larger than 32 MB. You may then create an extended partition which is divided into logical drives, 32–MB or smaller. These logical drives, however, are not bootable. Note that you may not combine method three with method one or two.

Select the method you would like to use.

If you choose Option 1 or 2, the following screen appears.



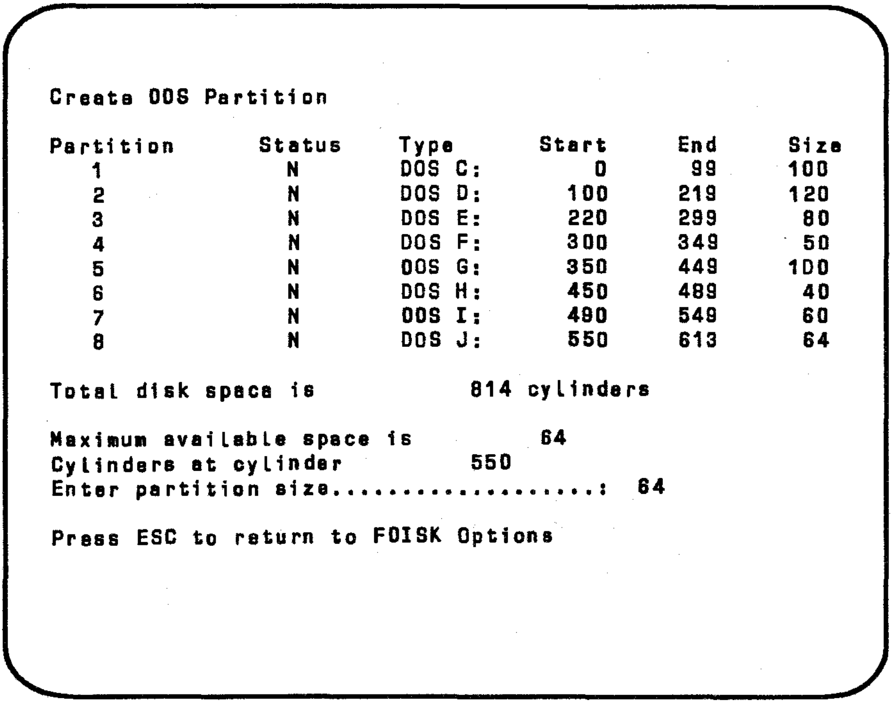
```
Create DOS partition

No partitions defined.

Total disk space is 614 cylinders

Maximum available space is    614
cylinders at cylinder       0
Enter partition size............:  [614]


Press ESC to return to FDISK Options
```

The default partition size will be adjusted to 32 MB for Option 2, or the maximum available size for Option 1.

Enter a number equal to the size in cylinders of the partition you want to create. If you enter **100**, FDISK creates a partition of 100 cylinders.

To create more DOS partitions, enter another partition size and starting cylinder number. FDISK keeps track of the disk space and limits the maximum partition size you can assign.

The following sample screen displays data for a hard disk with eight MS-DOS partitions.



```
Create DOS Partition

Partition       Status     Type       Start      End      Size
   1              N        DOS C:         0       99       100
   2              N        DOS D:       100      219       120
   3              N        DOS E:       220      299        80
   4              N        DOS F:       300      349        50
   5              N        DOS G:       350      449       100
   6              N        DOS H:       450      489        40
   7              N        DOS I:       490      549        60
   8              N        DOS J:       550      613        64

Total disk space is             814 cylinders

Maximum available space is           64
Cylinders at cylinder           550
Enter partition size...................:   64

Press ESC to return to FDISK Options
```

FDISK provides the following partition information:

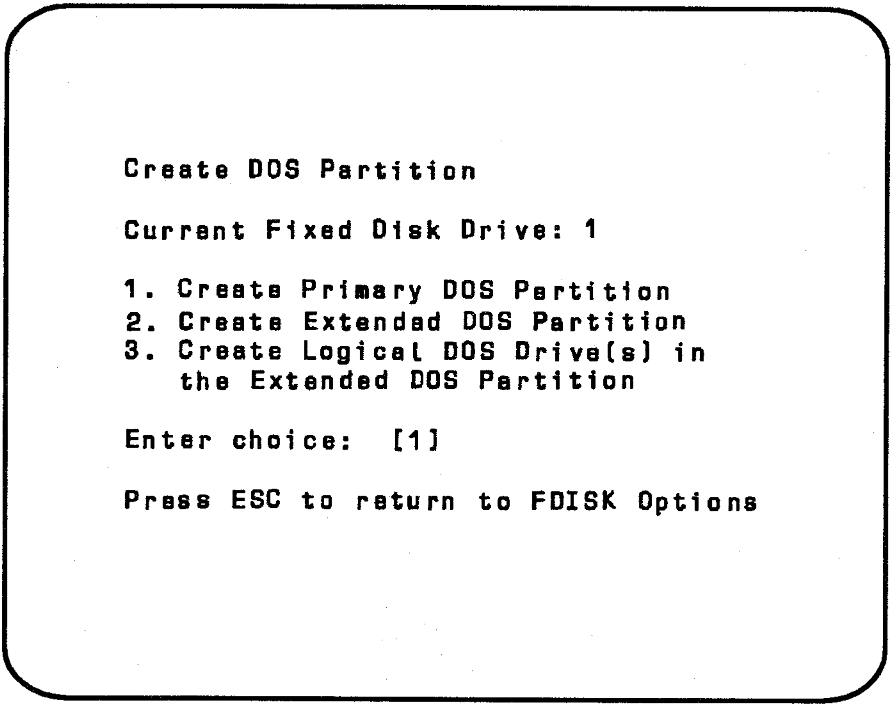
- The Partition column lists the number MS–DOS assigns to each partition.
- The Status column lists partition status: A for active and N for non-active.
- The Type column lists partition classification, MS–DOS or non–MS–DOS, and letters to distinguish MS–DOS partitions.
- The Start and End columns list cylinder numbers at the beginning and end of the partition.
- The Size column lists the number of cylinders that each partition contains.

This FDISK screen also displays the information that FDISK used to create Partition 8:

- Partition size: 64

- Starting Cylinder Number: 550

If you choose the third option on the Create Menu, and if your hard disk is not yet completely partitioned, FDISK displays a screen like the following.



```
Create DOS Partition

Current Fixed Disk Drive: 1

1. Create Primary DOS Partition
2. Create Extended DOS Partition
3. Create Logical DOS Drive(s) in
   the Extended DOS Partition

Enter choice:  [1]

Press ESC to return to FDISK Options
```

If no extended partitions exist, the third option is not displayed.

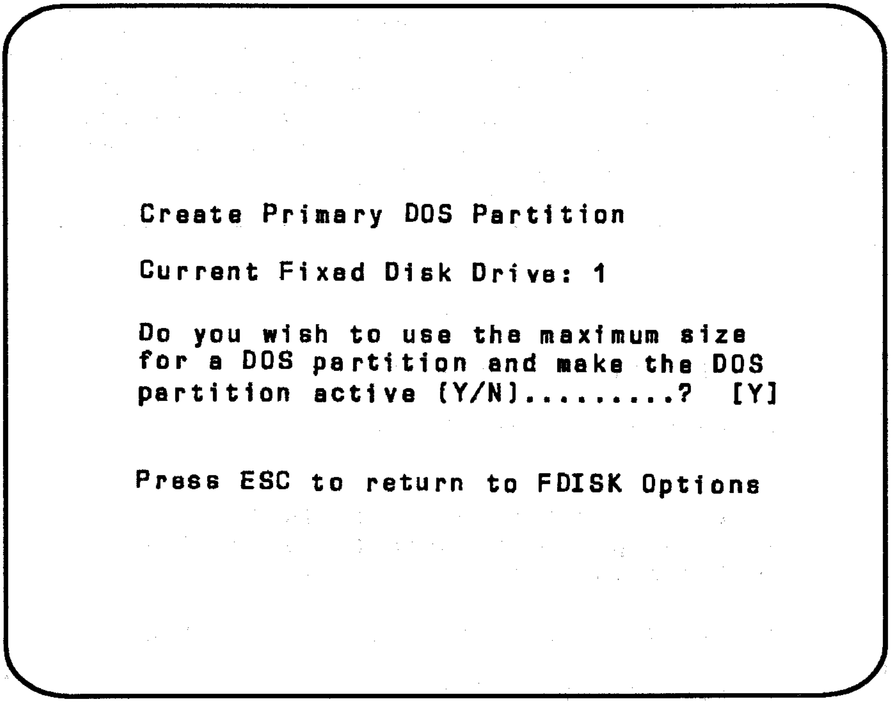
Selection 1: Create Primary DOS Partition

You must create a primary MS–DOS partition first before you can create any extended MS–DOS partitions on your disk. In most cases, you will need only one MS–DOS partition for your entire disk.

Creating a Primary MS–DOS Partition

To create a primary MS–DOS partition, press the **Enter** key to accept the primary MS–DOS default selection (1).

The Create Primary DOS Partition Menu appears next:



```
Create Primary DOS Partition

Current Fixed Disk Drive: 1

Do you wish to use the maximum size
for a DOS partition and make the DOS
partition active (Y/N).........?  [Y]


Press ESC to return to FDISK Options
```

Creating a Single MS–DOS Partition

If you use your entire hard disk for MS–DOS, you will use the FDISK program only one to create the primary partition. If you want to use the entire hard disk (up to 32 megabytes) for MS–DOS, press the **Enter** key to accept the default selection (Y). FDISK then displays the following message:

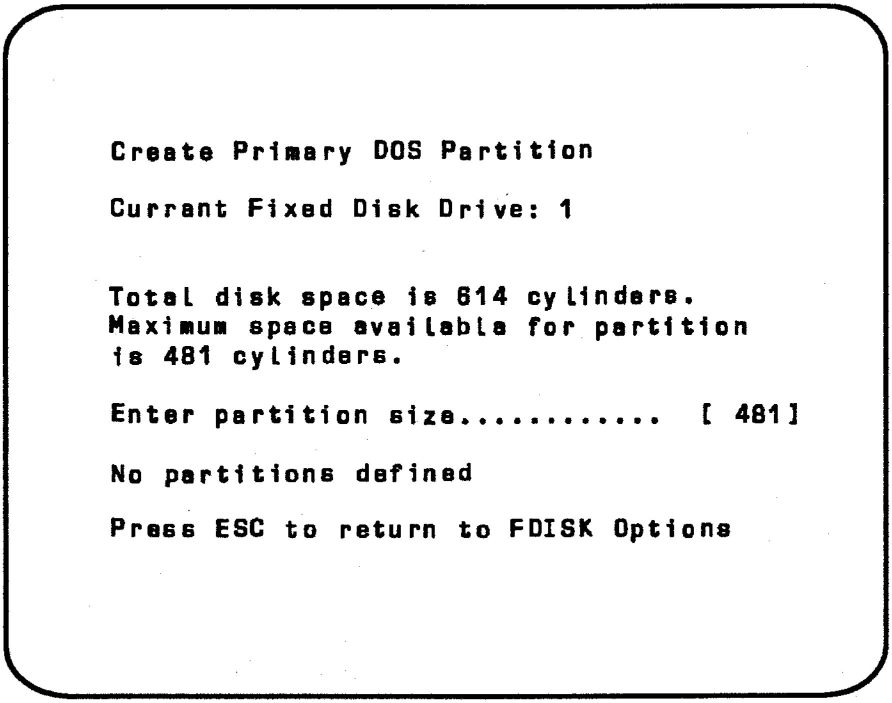
```
System will now restart

Insert DOS diskette in drive A:
Press any key when ready ...
```

Put your MS–DOS disk in drive A and press any key to restart MS–DOS.

```
FORMAT c: /s
```

Creating More Than One MS-DOS Partition

You may choose to create a primary MS-DOS partition smaller than the maximum size. To do this, type **N** (for No) in response to the question on the first Create Primary DOS Partition menu. FDISK displays a second Create Primary DOS Partition menu like the following. From this menu, you can specify the size of the primary MS-DOS partition:



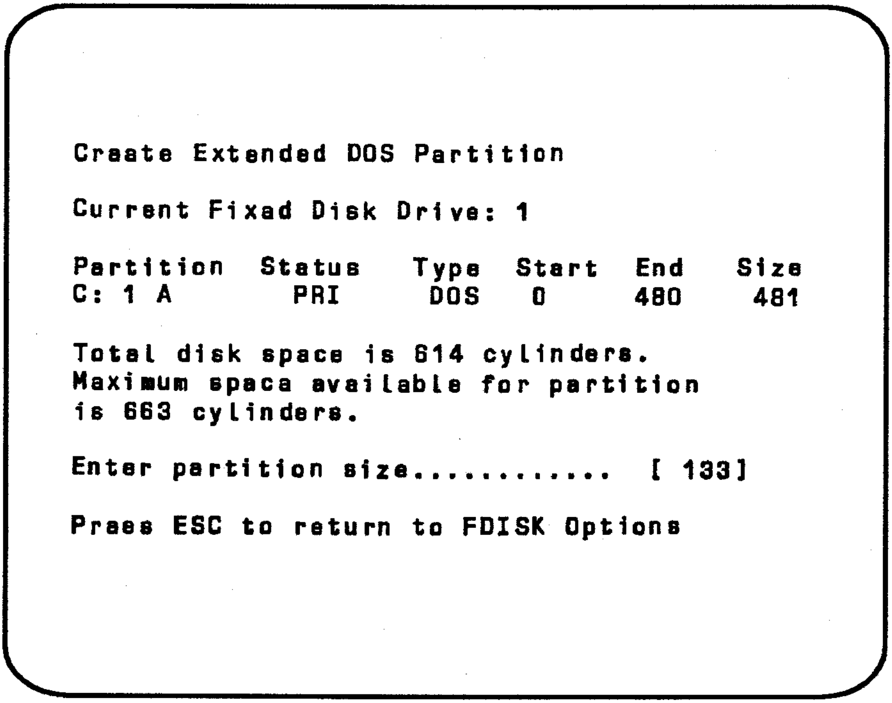
```
Create Primary DOS Partition

Current Fixed Disk Drive: 1


Total disk space is 614 cylinders.
Maximum space available for partition
is 481 cylinders.

Enter partition size............  [ 481]

No partitions defined

Press ESC to return to FDISK Options
```

The space available on your hard disk is measured in cylinders, also called tracks. This menu shows the total number of cylinders available for a hard disk partition, and prompts you to enter the size of your new partition. The default size for the partition is the maximum available space on the hard disk. Press the **Enter** key if you want the default size; otherwise, type the size (in cylinders) that you want for the partition, and press the **Enter** key. Any part of the disk that you do not use for the primary MS–DOS partition may be used for an extended MS–DOS partition.

Selection 2: Create Extended DOS Partition

You can use FDISK to create an extended partition if your disk is larger than 32 megabytes (the maximum partition size), or if you want to designate one or more logical drives for the disk.

To select Create Extended DOS Partition, type 2, and press the **Enter** key. In partition response, FDISK displays a menu like this one:



```
Create Extended DOS Partition

Current Fixed Disk Drive: 1

Partition  Status   Type  Start  End   Size
C: 1 A       PRI     DOS  0      480    481

Total disk space is 614 cylinders.
Maximum space available for partition
is 663 cylinders.

Enter partition size...........  [ 133]

Press ESC to return to FDISK Options
```

This menu shows the total number of cylinders available for an extended partition. The default for the partition size is the maximum available space on the hard disk. Press the **Enter** key if you want the default; otherwise, type the size (in cylinders) that you want for the partition, and press the **Enter** key.

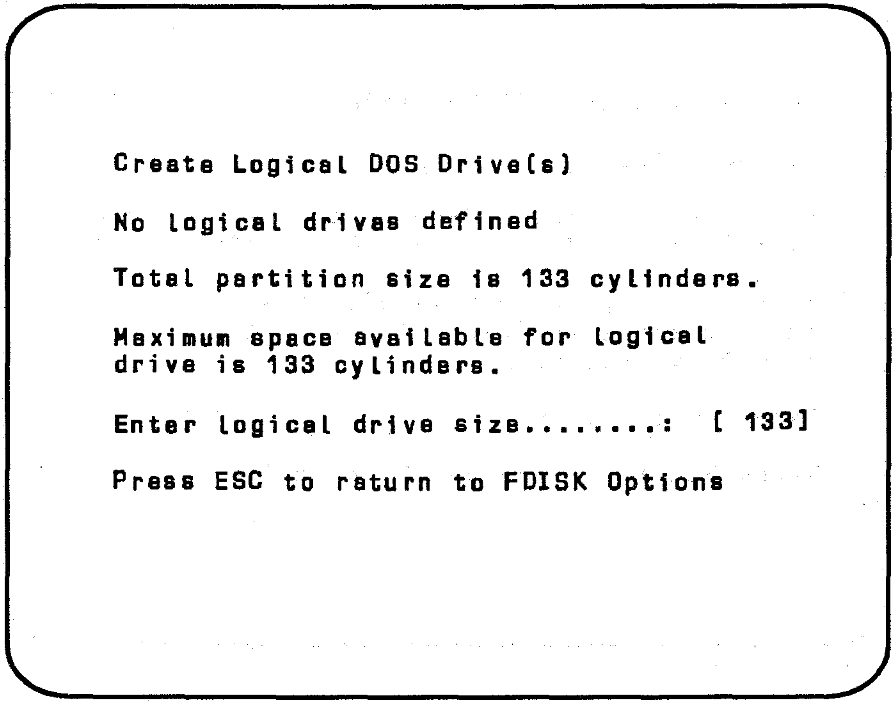
NOTE

> If FDISK finds any defective tracks at the start of the partition, it adjusts the partition boundaries to avoid those bad tracks.

Selection 3: Create Logical Drive in the Extended DOS Partition

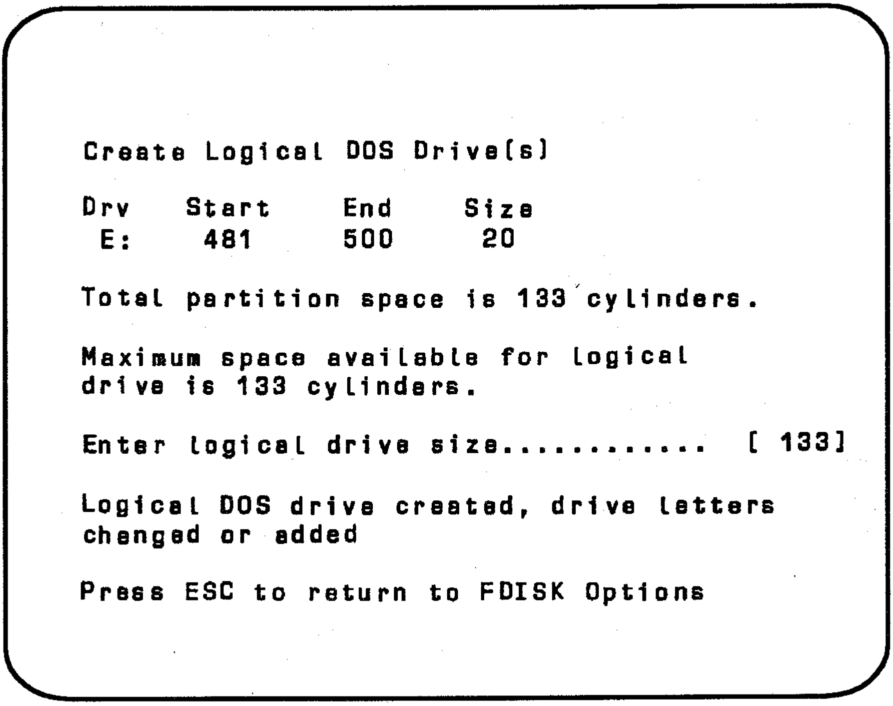
Creating Logical Drives in Extended Partitions

When you have created an extended partition, you must specify one or more drive letters that designate an area of the disk. A menu similar to the following is displayed if you choose option 3 from the Create DOS Partition Menu:



```
Create Logical DOS Drive(s)

No logical drives defined

Total partition size is 133 cylinders.

Maximum space available for logical
drive is 133 cylinders.

Enter logical drive size........:  [ 133]

Press ESC to return to FDISK Options
```

You may designate the entire partition as one logical drive, or divide it into two or more logical drives. For example, if you want to segregate a particular application and its data files to their own drive, you may want to create a second logical drive on the partition.



```
Create Logical DOS Drive(s)

Drv    Start     End     Size
 E:     481      500      20

Total partition space is 133 cylinders.

Maximum space available for logical
drive is 133 cylinders.

Enter logical drive size............  [ 133]

Logical DOS drive created, drive letters
changed or added

Press ESC to return to FDISK Options
```

Because you cannot use an MS-DOS extended partition without a drive letter, FDISK continues to prompt you for logical disk drive information until the whole partition has been assigned to a logical drive.

When the entire partition is assigned to logical drives, FDISK displays this message:

> All available space in the Extended DOS
> partition is assigned to logical drives.

Press **ESC** to return to the Main FDISK Menu.

NOTE

> FDISK automatically adjusts the starting point for a partition to the first good area on your disk. This adjustment can reduce the size of your partition.

After creating a DOS partition, you must format the partition. See the description of the FORMAT command for details.

If you want to boot (start MS-DOS) from this MS-DOS partition, you must

- Format the partition using the FORMAT command with the /S option.

  (For more information about the FORMAT command, see Section 4, "MS-DOS Commands.")

  Note that only NEC partitions (compatible and large) or MS-DOS primary partitions may be bootable.

- Change the active partition.

OPTION 2: CHANGING THE ACTIVE PARTITION

Option 2 allows you to change the active partition (i.e., the partition that controls your computer when you turn it on or reboot it).

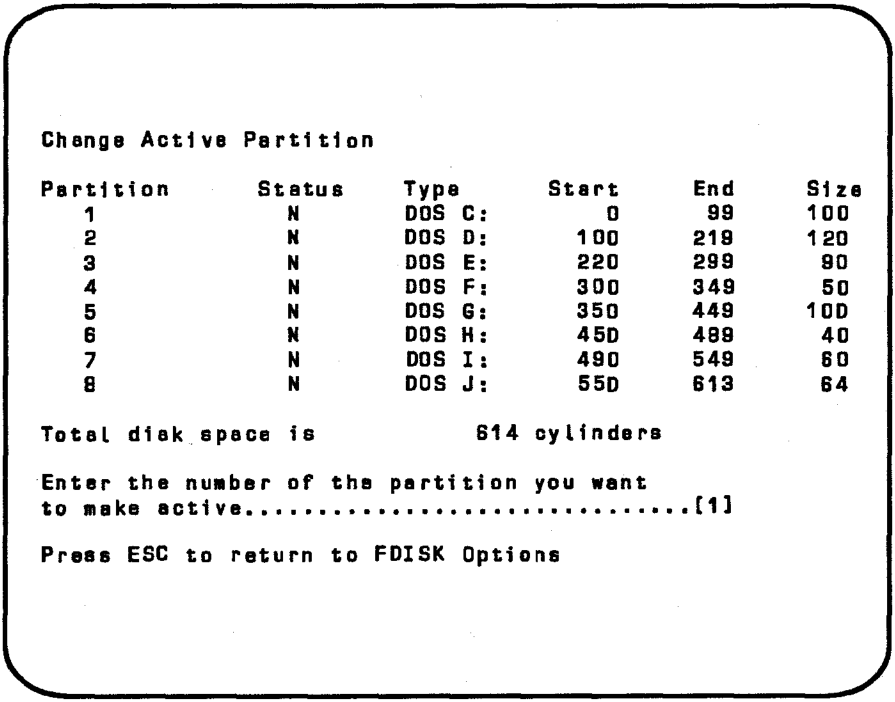
NOTE

> Extended partitions cannot be made active.

Only one partition can be active at a time on a hard disk.

To change the active partition, type **2** in the FDISK Options Menu. FDISK then displays the Change Active Partition screen.

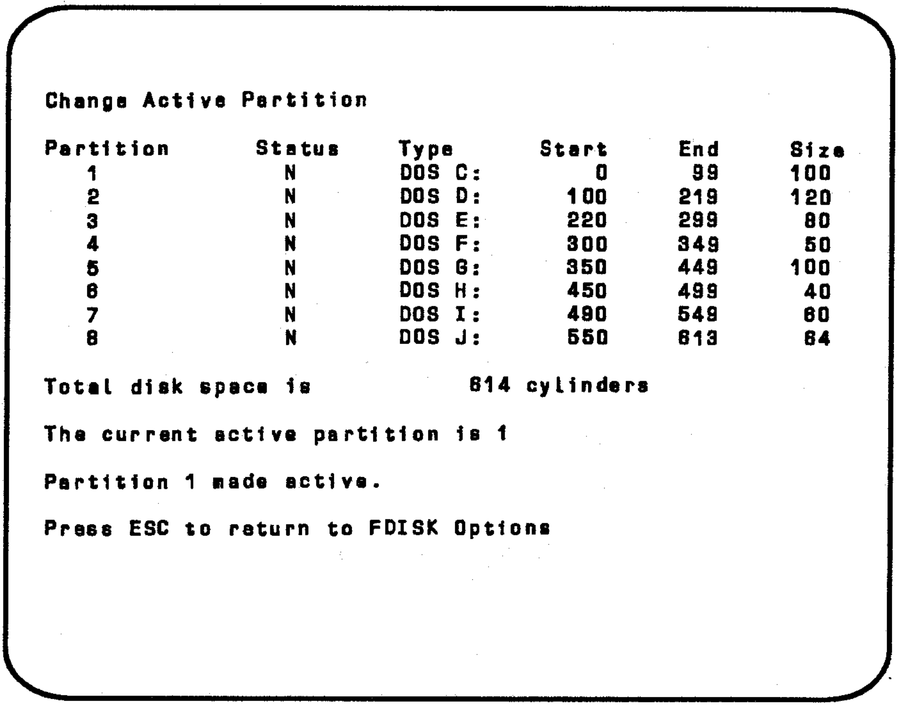
The following sample screen shows a hard disk with eight nonactive partitions.



```
Change Active Partition

Partition       Status      Type       Start       End       Size
   1              N         DOS C:         0        99        100
   2              N         DOS D:       100       219        120
   3              N         DOS E:       220       299         90
   4              N         DOS F:       300       349         50
   5              N         DOS G:       350       449        100
   6              N         DOS H:       450       489         40
   7              N         DOS I:       490       549         60
   8              N         DOS J:       550       613         64

Total disk space is            614 cylinders

Enter the number of the partition you want
to make active..............................[1]

Press ESC to return to FDISK Options
```

Change the active partition by typing the partition number that you want to activate and pressing **Enter**. You can activate partition 1, the default, by simply pressing **Enter**.

If you activate partition 1, a screen similar to the following appears.



```
Change Active Partition

Partition          Status      Type        Start       End       Size
    1                N         DOS C:          0        99       100
    2                N         DOS D:        100       219       120
    3                N         DOS E:        220       299        80
    4                N         DOS F:        300       349        50
    5                N         DOS G:        350       449       100
    6                N         DOS H:        450       499        40
    7                N         DOS I:        490       549        60
    8                N         DOS J:        550       613        64

Total disk space is            614 cylinders

The current active partition is 1

Partition 1 made active.

Press ESC to return to FDISK Options
```

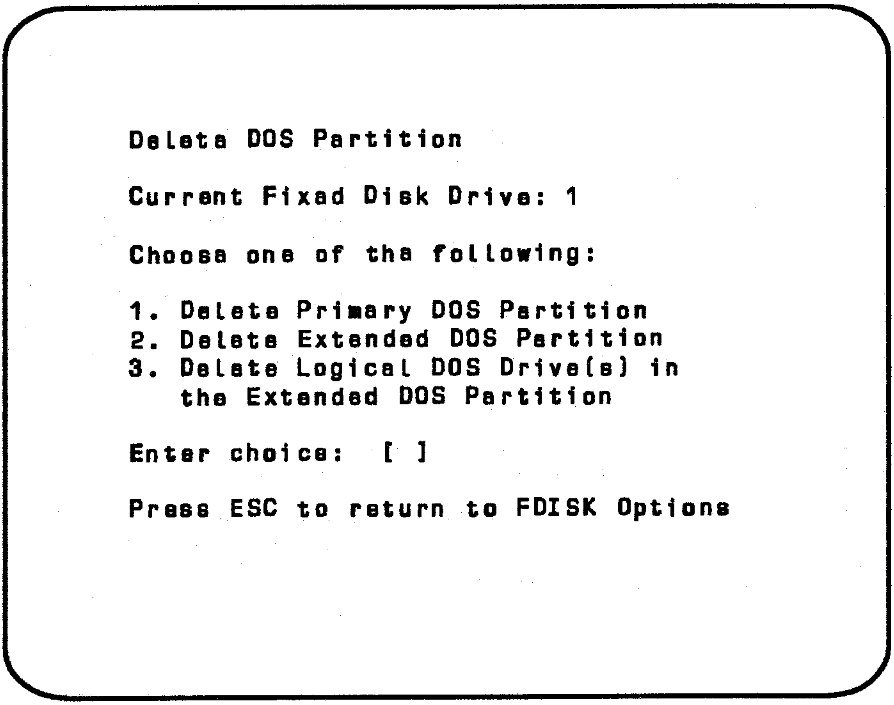
Once you change the active partition, you must reboot the system.

OPTION 3: DELETING A DOS PARTITION

When you delete a DOS partition, FDISK deletes the partition boundaries and any data that existed in that partition. Once you delete the partition, you cannot recover the data that was on it.

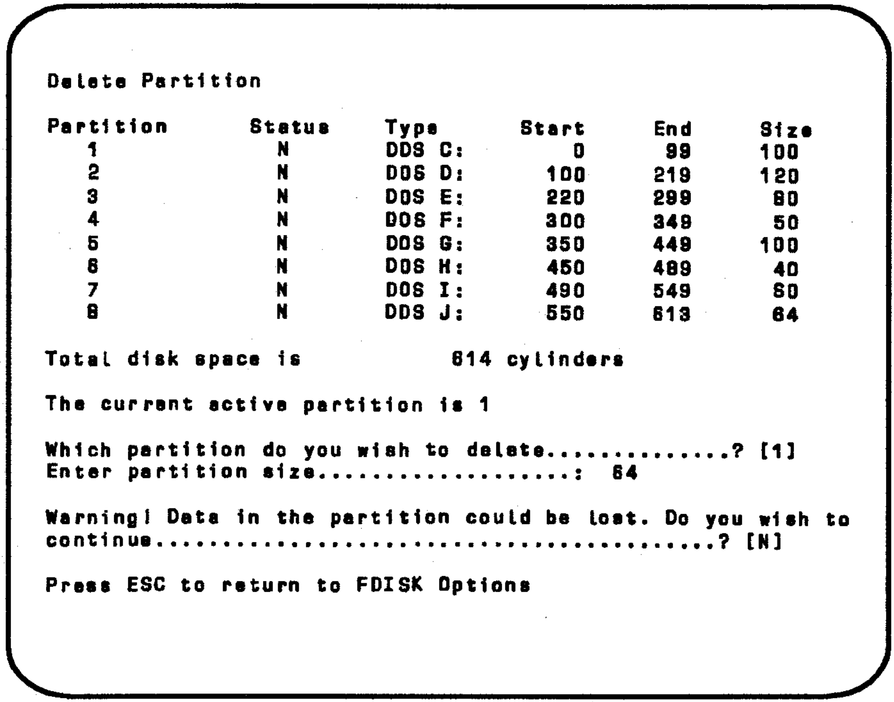
Note that you cannot use FDISK to delete a non-DOS partition. Instead, to continue using MS-DOS after you have deleted the DOS partition, you must put a MS-DOS program disk into drive A. To start a different operating system in another partition of your hard disk, you must change the active partition to that number before you delete the DOS partition.

If you choose the third option on the main menu, FDISK displays the following menu, which asks you to identify whether the partition you want to delete is a primary or extended DOS partition. If no logical drives are defined in the extended DOS partition, selection 3 does not appear.



```
Delete DOS Partition

Current Fixed Disk Drive: 1

Choose one of the following:

1. Delete Primary DOS Partition
2. Delete Extended DOS Partition
3. Delete Logical DOS Drive(s) in
   the Extended DOS Partition

Enter choice:  [ ]

Press ESC to return to FDISK Options
```

Type the number of the selection you want and press the **Enter** key. The next menu, whether for a primary or extended DOS partition, shows the status of that partition. Selection 1 allows you to delete an NEC partition or a DOS partition.



```
Delete Partition

Partition        Status      Type        Start      End       Size
    1              N         DOS C:          0       99        100
    2              N         DOS D:        100      219        120
    3              N         DOS E:        220      299         80
    4              N         DOS F:        300      349         50
    5              N         DOS G:        350      449        100
    6              N         DOS H:        450      489         40
    7              N         DOS I:        490      549         60
    8              N         DOS J:        550      613         64

Total disk space is           614 cylinders

The current active partition is 1

Which partition do you wish to delete..............? [1]
Enter partition size...................:  64

Warning! Data in the partition could be lost. Do you wish to
continue.......................................? [N]

Press ESC to return to FDISK Options
```

Delete an NEC or Primary DOS partition by typing the appropriate number in response to the question "Which partition do you wish to delete....?"

FDISK then warns you that the data in the partition could be lost and prompts you as follows:

> Do you wish to continue......? [y]

Enter **Y** to instruct FDISK to delete the partition you specified.

For example, if you enter **3** and Y, FDISK deletes partition 3 (cylinders 220–299).

Deleting an Extended DOS Partition

If you choose to delete an extended partition, you must first delete the logical drives associated with that partition.

Selection 3: Delete Logical Drive in the Extended DOS Partition

Deleting a Logical Drive

To delete a logical drive, type **3** to select the option Delete Logical DOS Drive(s) in the Extended DOS Partition from the Delete DOS Partition menu. Then press the **Enter** key. FDIWK displays a list of logical drives on the extended partition.

Type the letter of the drive you want to delete, and press the **Enter** key. FDISK displays this message:

```
Are you sure ............................ ? [N]
```

If this logical drive contains valuable data you have not backed up, press the **Enter** key. This stops FDISK from deleting the logical drive.

NOTE

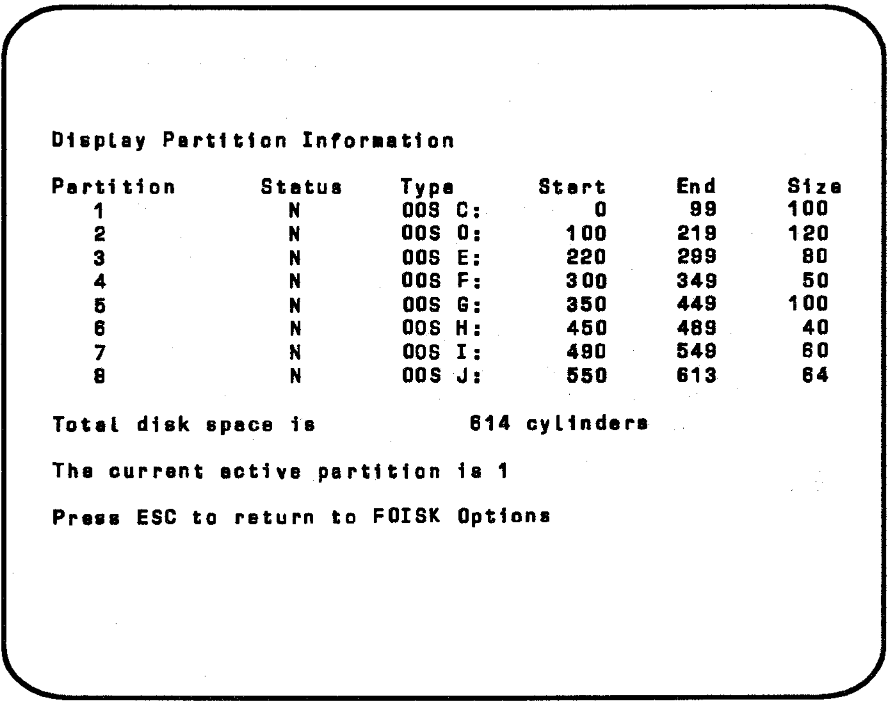
> Be sure to back up all files you will need from the logical drive before you delete the drive. When FDISK deletes a logical drive or partition, the data is destroyed.

If you are sure you want to delete the drive, type **Y** (for Yes).

OPTION 4: DISPLAYING PARTITION INFORMATION

Option 4 permits you to list information on all hard disk partitions.

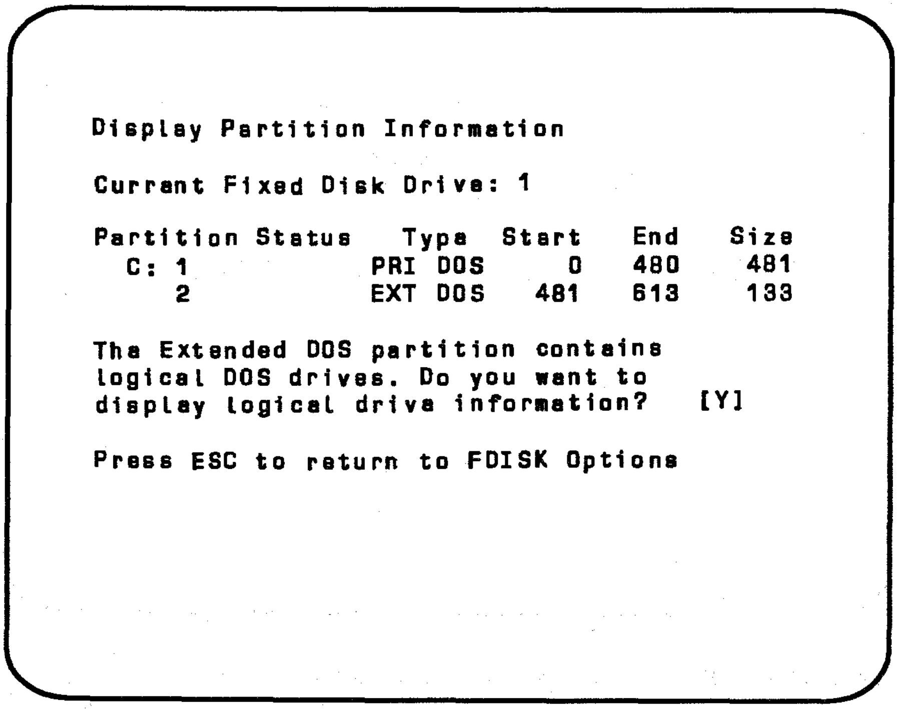
To display partition information, type **4** in the FDISK Options menu. A display partition screen similar to the following appears.



```
Display Partition Information

Partition        Status     Type       Start      End       Size
    1              N        OOS C:         0       99       100
    2              N        OOS O:       100      219       120
    3              N        OOS E:       220      299        80
    4              N        OOS F:       300      349        50
    5              N        OOS G:       350      449       100
    6              N        OOS H:       450      489        40
    7              N        OOS I:       490      549        60
    8              N        OOS J:       550      613        64

Total disk space is          614 cylinders

The current active partition is 1

Press ESC to return to FOISK Options
```

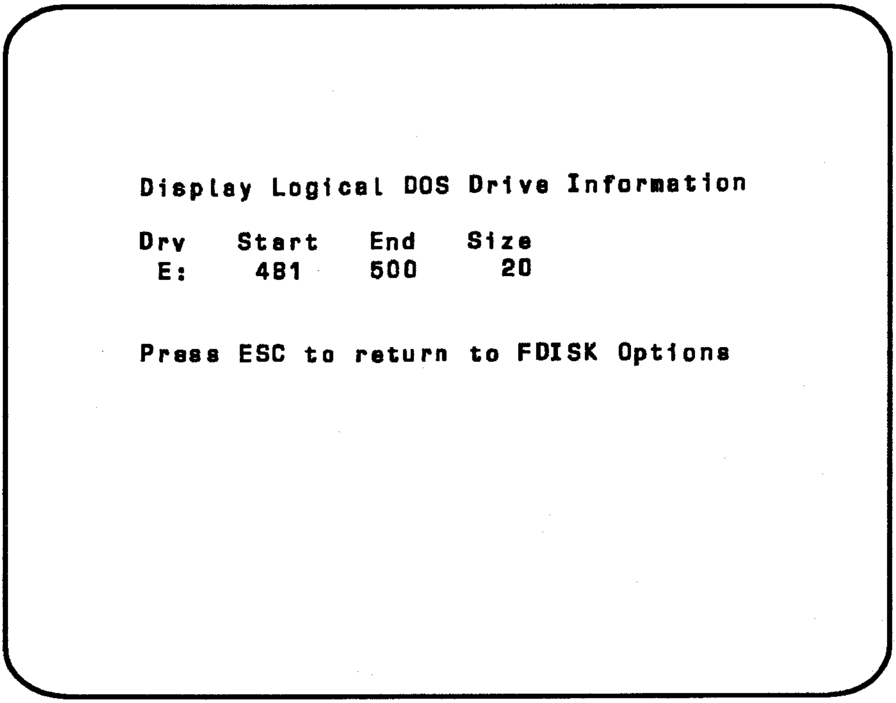
The Partition Information indicates that FDISK has removed partition 3 (cylinders 220–299 are missing) and partition 1 is active.

If you have an extended partition with logical drives, the following screen will appear.



```
Display Partition Information

Current Fixed Disk Drive: 1

Partition Status   Type  Start   End   Size
  C: 1            PRI DOS     0   480   481
     2            EXT DOS   481   613   133

The Extended DOS partition contains
logical DOS drives. Do you want to
display logical drive information?   [Y]

Press ESC to return to FDISK Options
```

Press **Y** to display the logical drives in the extended partition.



```
Display Logical DOS Drive Information

Drv    Start   End    Size
 E:     481    500      20


Press ESC to return to FDISK Options
```

OPTION 5: SELECT THE NEXT HARD DISK DRIVE

Option 5 allows you to set up a second hard disk, if your system has two hard disks.

To prepare a second hard disk, type **5** in the FDISK Options menu.

Once you finish partitioning your hard disk, you can format the disk and then copy MS-DOS system commands to the partition.

Index