PC2086

User Manual and MS-DOS® Guide

AMSTRAD

IMPORTANT SAFETY NOTICE:

When choosing a position for the PC, make sure that all the equipment is installed close to, and within easy access of, the electrical mains supply socket.

The product described in this manual and products for use with it are subject to continuous development and improvement.

This manual is provided to you free of charge and is intended only to assist the reader in the use of the product and although the information contained in this manual and literature provided with the product is given by AMSTRAD in good faith, AMSTRAD does not warrant the accuracy of that information and accepts no responsiblity or liability for any loss or damage arising from the use of any information provided or omitted.

You must carefully read these instructions and all other literature provided with the product, including the Guarantee Card, prior to operation.

IMPORTANT
The terms and instructions, contained in the Guarantee Card provided with the product, must be strictly complied with.

EXCLUSIONS OF CONSEQUENTIAL LOSS
IN ANY EVENT, AMSTRAD ACCEPTS NO LIABILITY FOR ANY CONSEQUENTIAL LOSS OR DAMAGE ARISING FROM THE USE OR FAILURE OF THE PRODUCT OR ANY INFORMATION PROVIDED, INCLUDING, BUT NOT LIMITED TO, ECONOMIC OR FINANCIAL LOSS, DAMAGE TO PERIPHERAL EQUIPMENT OR PRODUCTS, LOSS OF USE, PRODUCTIVITY OR TIME.

All correspondence relating to the product or this manual should be addressed to:

AMSTRAD PLC
Brentwood House
169, Kings Road
BRENTWOOD
Essex CM14 4EF

Written by Susan Vass and Jean Gilmour with contributions from the Amstrad Technical Department

Published by AMSTRAD PLC
First published 1986
Revised for PC1640 1987
Revised for PC2086 1988

PREFACE

This manual is in two parts:

Part 1 - Guide to using the PC

Part 2 - Guide to using MS-DOS (Microsoft® Disk Operating System).

Each chapter, in Part 1, gives an introductory description of a part, or parts of the PC: how each part works and what it can be used for. Highly technical information is set out at the end of each chapter. This is required for writing programs and performing specialised tasks with the PC.

Do not worry if you do not understand the technical information; this would indicate that you do not need to use it. The majority of users most certainly will not need to understand this information.

The descriptions of the various parts of the PC contain information about how each part works and what it can be used for.

The first thing you will probably want to do is install the PC, rather than read the manual! Chapter 3 describes the installation steps, so if you feel confident that you know how you want to set-up your PC you could start installing straight away.

If you are new to computing you may want to understand what you are doing during the installation process, in which case, you could read Chapter 2 first, which gives a broad view of the possible uses of the PC.

If you run into difficulties, you should look at Part 2, Chapter 10 "Troubleshooting", and see if you can find out what has gone wrong, and what to do about it. If you experience difficulties which are not covered in this manual, consult your dealer.

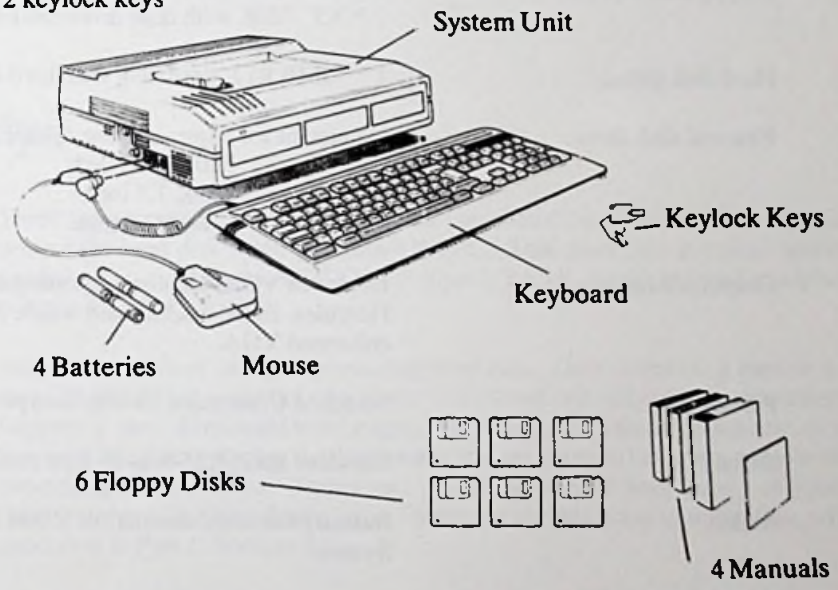
PART ONE - CONTENTS

1. REFERENCE INFORMATION

1.1 Contents of the Package

Check the contents of the package

- A PC System Unit, with built-in disk drives in the front panel
- A Keyboard
- A Mouse
- Four AA size Batteries (slotted into the System Unit packaging)
- Four manuals
- Six 3.5 inch Floppy Disks
- 2 keylock keys



If any item is missing, consult your dealer.

Save all the packing materials in case you want to transport your PC at some later date.

1.2 Specification

The specification details will be required if you decide to buy Application programs to use on your PC, and additional hardware, like a printer, or a display or expansion cards, for example.

Built-in standard features

Memory size - RAM:	640 Kilobytes
Central Processing Unit (CPU):	8MHz 8086 processor
Keyboard:	102 key Enhanced Keyboard
Security:	keylock switch
Mouse interface:	
Floppy disk drive(s):	1 * 3.5" 720K with single and hard disk models, 2 * 3.5" 720K with dual drive model
Hard disk drive:	1 * 32MB RLL hard disk with hard disk model
External disk drive:	Connector for external drive. Either 360K 5.25 inch 720K 3.5 inch 1.44M 3.5 inch
Graphics Adaptor:	Extended VGA adaptor supporting MDA, CGA Hercules, EGA, MCGA and VGA. Also supports enhanced VGA.
Parallel Port:	Standard Centronics 25-way D-type connector.
Serial Port:	Standard RS232 25-way D-type connector.
BIOS:	Industry standard compatible ROM Operating System
Expansion slots:	3 full size 8 bit expansion slots
Operating System:	MS-DOS 3.30
Additional Software:	Microsoft Windows Microsoft GW-BASIC.

2. GUIDE TO USING THE PC FOR NEW USERS

Introduction

This chapter gives a broad view of the activities which are fundamental to the PC - safety, and using an operating system and of the activities which are optional - using a graphic interface, programming languages, application programs, and additional hardware.

It also tells you where to find the instructions for all these activities.

2.1 Safety

There are two main aspects of safety that you should be aware of. One of which is to protect the hard disk when you move the PC. This procedure is called "parking the heads". You can use the MS-DOS command, PARK, which is listed in alphabetical order in Part 2, Chapter 8.

The other aspect of safety is protecting your data. Data stored on a hard or a floppy disk can be lost for a number of reasons; it is almost certainly very inconvenient if this happens to you. You could lose programs and data which are irreplaceable, or at least, you may find that replacing the data means you are going to have to re-type or re-install everything you have lost. Therefore, you are advised to keep copies of your stored data, on a hard disk or a floppy disk. There are various ways of doing this, which are described in Part 1, Section 7.6.

2.2 The Operating System (MS-DOS)

If you are new to computing, you will need to understand what an operating system is.

The computer is a collection of advanced electronics which cannot work without being told what to do. It needs an operating system to control it and to tell it what to do.

The operating system supplied with your PC is called MS-DOS (Microsoft® Disk Operating System). You issue instructions, called commands, to the operating system, and it, in turn, controls the electronics of the PC.

Once it is told what to do, the operating system can make the computer perform functions like running programs, storing and retrieving data from files on disks, as well as controlling smaller operations like accepting characters from the keyboard and displaying them on the screen.

You could use an operating system other than MS-DOS. If you have a hard disk you could re-configure the disk and install one or more operating systems. Details of re-configuring the hard disk are to be found in Chapter 7. If you have a floppy disk drive PC, you can load an alternative operating system from a floppy disk.

2.3 Graphic Interface (Windows)

A graphic user interface displays the operating system's activities, and the contents of your disks in a diagrammatic and pictorial form.

A graphic form of display makes it easier to issue commands; a mouse is used to control a pointer which you can move around the screen to select items and operate menus. See Chapters 4 and 6 for more information, and your Windows User Guide.

A graphic interface, such as Windows, which is supplied with your PC, has to be used in conjunction with the appropriate operating system. Windows is written to supplement MS-DOS - it is an addition to the operating system. If you want to use word processing programs, spreadsheets or a database, you will need to install a third level of programs onto your PC. These programs are called Application progams, examples of which are: Wordstar, Supercalc, and Infomaster.

Once you have learned how to use a graphics interface, you will realise the advantage of being able to use a number of application programs which work with the same

interface. It means that you will not have to learn how to control each application program; you will control the interface and it will control the programs.

The PC is supplied with the Microsoft Windows Presentation Manager which is one such graphics interface. The system of menus and controls used in Windows conforms to the Common User Access of IBM's Systems Application Architecture (SAA) which is fast becoming the industry standard for graphics interfaces.

Windows also brings a secondary benefit in that it provides multi-tasking so it is possible to run more than one program at once - this is not even limited to programs written specifically to work with Windows. It means, for example, that you can be running both a spread-sheet and word processor at the same time and can switch between the two to exchange data. For more information about multi-tasking, see Part 1, Section 7.2

2.4 Application Programs

The primary use for your PC will possibly be to run application programs such as word processors, databases, spreadsheets, accounting and invoicing systems and so on.

Because the PC is compatible with the IBM PC there already exists a huge range of software to choose from, virtually all of which will run on the PC.

When buying application programs, check the specification details of the PC, in Chapter 1, against the specification details of the programs, or consult your dealer.

The PC supports most of the common graphics modes so increasing the choice of software that you could use on it.

2.5 Programming Languages (GW-BASIC)

You may be interested in developing your own PC applications if the type you want is not available. There are many computer languages available that allow you to formulate a series of instructions that make the PC do a specific task.

The PC is supplied with Microsoft® GW-BASIC which is an industry standard implementation of one of the easiest to learn programming languages, that is, BASIC. GW-BASIC allows you to run programs written in IBM BASIC and IBM BASICA. For instructions on how to use this language, see your GW-BASIC manual.

2.6 Expanding the PC

The possibilities for expanding your PC are virtually limitless. The parallel and serial ports will allow you to connect almost any printer to the PC and in addition the serial port may also be used for connecting your PC to another computer, a terminal or modem allowing you to access many large computers via the telephone system.

The PC also has an external floppy disk drive connector, to which an additional floppy disk drive may be added. It can be used to increase the amount of storage capacity so that while programs are running, one drive can hold a disk which contains the programs and the second drive can hold the disk which contains the data.

Within the System Unit are three full size expansion slots which can take any of the cards designed to be used in the IBM PC. Expansion cards can contain various features: they could be used to provide more memory, a hard disk drive, further serial or parallel interfaces, or an internal modem, for example.

3. INSTALLATION

Setting-up the PC

Introduction

This chapter tells you how to set up and prepare your PC for future use. However experienced you are in using microcomputers, you are advised to follow these instructions: from step 1 to step 11. There are no shortcuts. If you rush on too quickly to running programs, you risk corrupting the software supplied with your PC. The consequence of this will be that you will have to buy new copies of this software from your dealer.

To set up your PC you will need a clear, dust-free area on a desk or table that is 4ft wide and 3ft deep.

If you do not have a hard disk PC, you will need at least one blank floppy disk. You will need two mains plugs, one for the System Unit and one for the Display.

Before you install the PC, make sure you have unpacked all the contents described in Chapter 1.

3.1 Installation steps

The Installation steps are as follows:

1. **Fit the plugs**
2. **Insert the batteries**
3. **Connect the Display to the System Unit**
4. **Connect the Keyboard and Mouse to the System Unit**
5. **Connect a printer to your PC if you have one**
6. **The startup procedure**
   - **Check that the hard disk PC is prepared for use**
7. **Load the supplied software: MS-DOS and Windows**
   - **Hard disk PC**
   - **Dual drive PC**
   - **Single drive PC**
8. **Set Time and date**
9. **Back-up your PC disks**
10. **How to reset your PC**
11. **How to switch off your PC properly**

Each step is described in detail in the following pages.

Step 1. Fit the plugs

The PC operates from a 220-240 Voit 50Hz AC mains supply.

Fit two proper mains plugs to the mains leads on the Display and the System Unit. If a 13 Amp (BS1363) plug is used, a 5 Amp fuse must be fitted. The 13 Amp fuse supplied in a new plug must not be used. If any other type of plug is used a 5 Amp fuse must be fitted in the plug or in the adaptor or at the distribution board.

WARNING - THIS APPARATUS MUST BE EARTHED

IMPORTANT:
The wires in the mains lead are coloured in accordance with the following code:

Green and Yellow	: Earth
Blue	: Neutral
Brown	: Live

As the colours of the wires in the mains lead of this apparatus may not correspond with the coloured markings identifying the terminals in your plug, proceed as follows:

i) The wire which is coloured green and yellow must be connected to the terminal in the plug which is marked by the letter 'E' or by the safety earth symbol ⏚ coloured green and yellow.

ii) The wire which is coloured blue must be connected to the terminal which is marked with the letter 'N' or coloured black.

iii) The wire which is coloured brown must be connected to the terminal which is marked with the letter 'L' or coloured red.

IMPORTANT:
DO NOT PLUG YOUR PC INTO THE MAINS SUPPLY YET

Always disconnect the mains plug from the supply socket when not in use.

Do not attempt to remove any screws or to open the casing of either the Display or the System Unit. Always obey the warning on the rating label on the back of the Display.

WARNING - LIVE PARTS INSIDE. DO NOT REMOVE ANY SCREWS.

WARNING

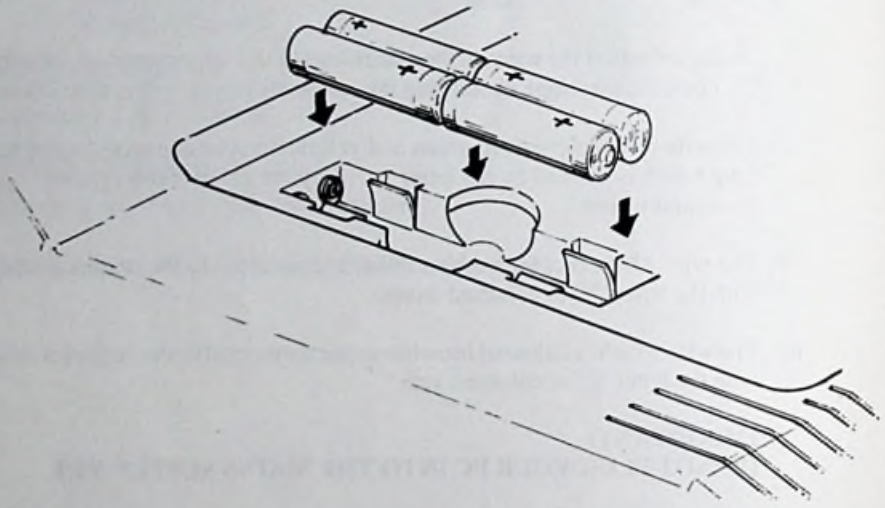
DO NOT CONNECT TO "I T" POWER SYSTEM

Note:
An "I T" power distribution system has no direct connection to earth; the exposed conductive parts of the electrical installation are earthed.

Step 2. Insert the batteries

Slot the four AA batteries into the battery compartment on the top of the System Unit as shown below.

Check that you put each battery the right way round, with its positive (+) end over the plus sign inside the battery housing.



The batteries are used to power the part of your PC's memory which stores information that needs to be carried over from one time you use your PC to another, for example, the setting of the date and time clock or the details of the floppy disk drives you have attached to your PC. This section of the memory is called the battery-backed RAM. It is constantly powered from the batteries to ensure this information is not lost.

These batteries will need changing about once a year. We recommend that you replace the batteries with new ones on a regular basis before the old batteries have gone flat.

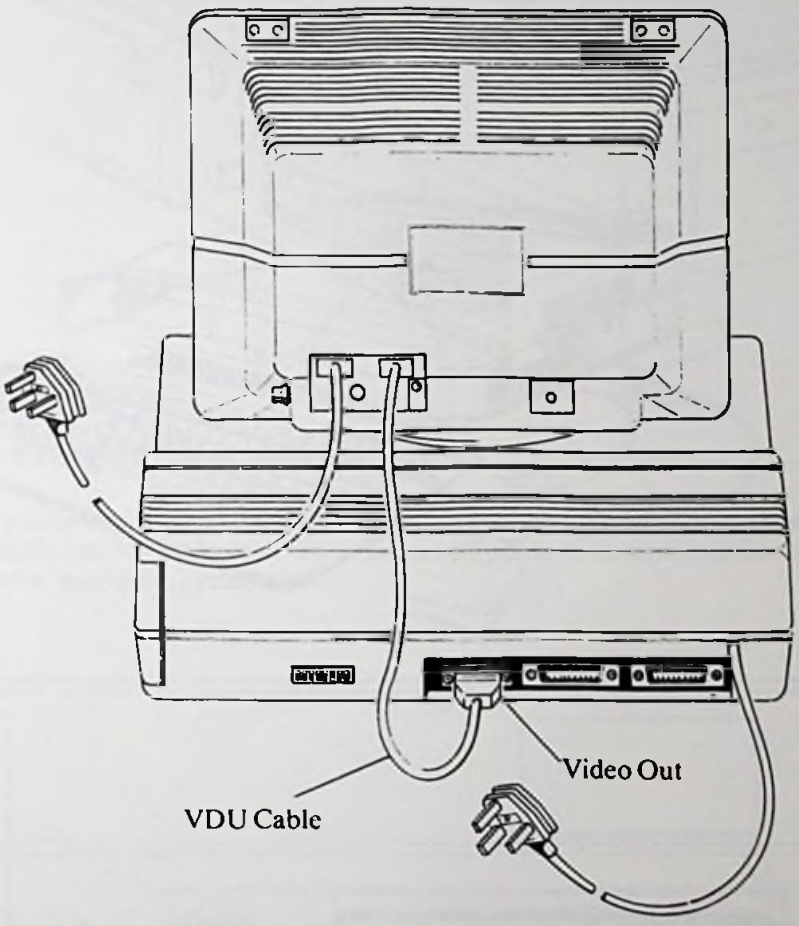
How to change the batteries is explained in Chapter 9, "Maintenance".

Step 3. Connect the Display to the System Unit

i) Check that the Display is not plugged into the mains supply.

ii) Place the display on top of the System Unit. The base of the Display fits into the recess in the top of the system unit.

iii) Insert the 15-pin plug on the end of the VDU cable leading from the back of the Display into the socket marked "video out" on the back panel of the System Unit.



iv) Arrange the System Unit and keyboard on your table, ready for use. Swivel the Display so that its screen is at a suitable angle. You can adjust this angle at any time.

Note You do not have to adjust any Display Selector switches before you use your PC.

Step 4. Connect the Keyboard and the Mouse to the System Unit and unlock the keylock

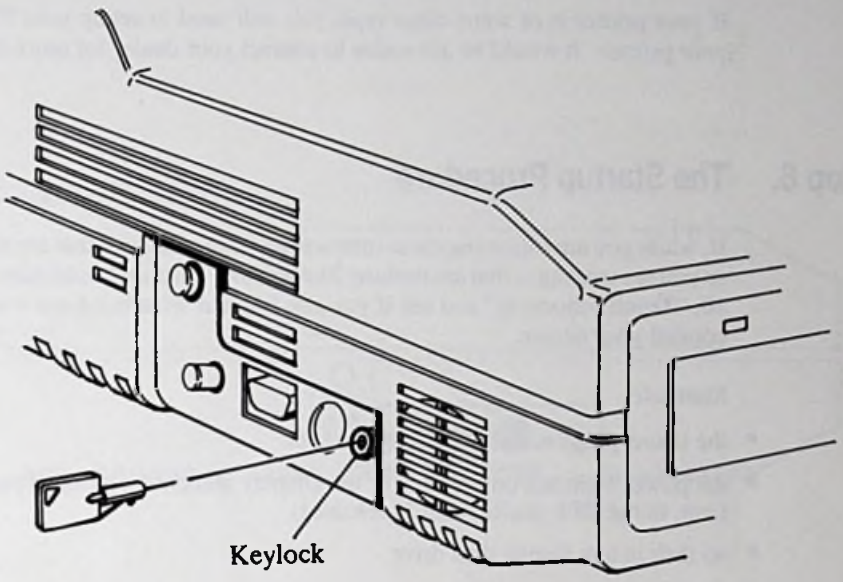
i) Check that your PC is not plugged into the mains supply

ii) Connect the keyboard to the system unit by inserting the right-angled plug at the end of the keyboard cable into the socket marked "Keyboard" on the left hand panel of the System Unit.

iii) Connect the mouse to the System Unit by inserting the 9-way 'D-type' plug on the end of the mouse lead into the socket marked "mouse" on the lefthand panel of the system unit.

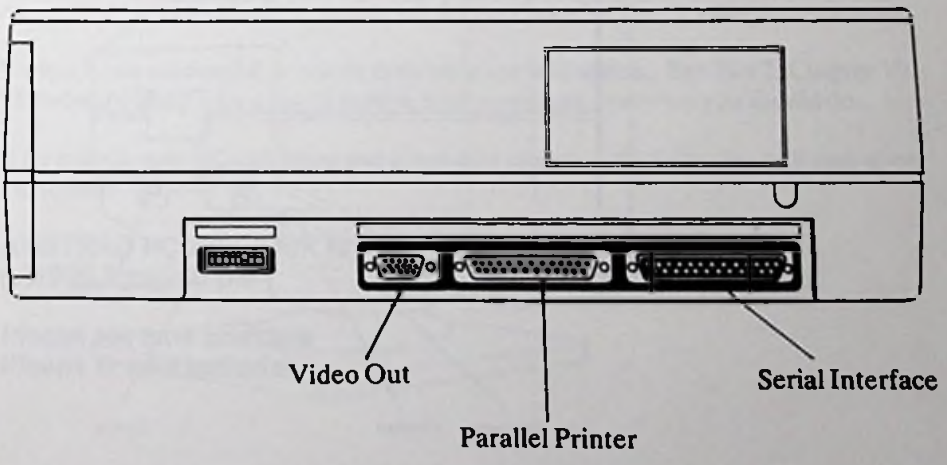
To use the mouse, refer to the description of the MOUSE command in Chapter 6.

iv) Unlock the keylock on the lefthand panel of the System Unit. While this switch is locked, the keyboard is disabled, the system will not start and anything typed on the keyboard will not be accepted by the system.



Step 5. Connect a printer to your PC

If you have a printer you may well want to connect this to your PC. If you have a serial printer, you will need a cable that connects your printer to the serial interface connector on the back of the System Unit.



If you have a parallel (or Centronics) printer, you will need a cable that connects your printer to the parallel printer connector on the back of the system unit. Your dealer will be able to tell you what type of cable you need.

If your printer is of some other type, you will need to set up your PC specifically for your printer. It would be advisable to contact your dealer for more information.

Step 6. The Startup Procedure

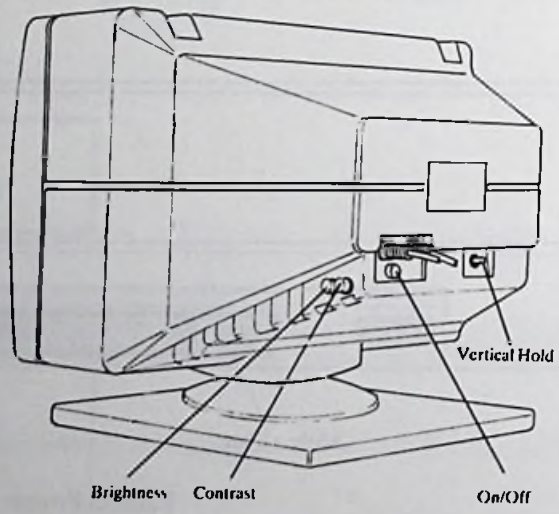
If, while you are following these instructions, things happen that are not explained here, or you see messages that are nothing like the ones described here, turn to Part 2, Chapter 10, "Troubleshooting" and see if you can find out what has gone wrong. If you can't, consult your dealer.

Start with

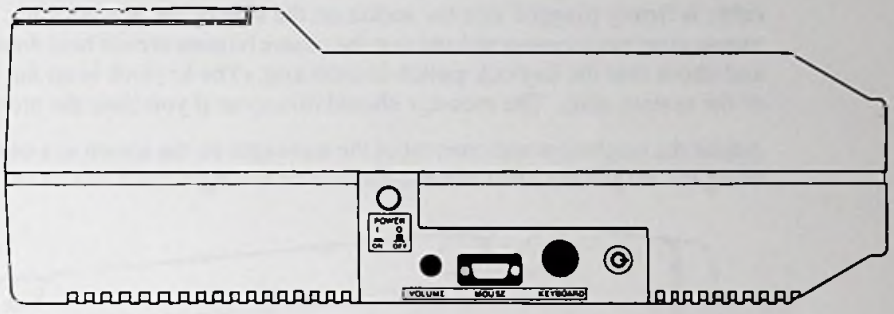
- the mains plugs out of the supply sockets
- the power switches on the back of the Display and on the lefthand panel of the System Unit, in the OFF position (fully released)
- no disk in any floppy disk drive

Then

- plug the Display into the mains supply
- plug the System Unit into the mains supply
- turn the machine on by pressing the power switches on the back panel of the Display....



...and on the lefthand panel of the System Unit.



The message **Please wait.....** should appear on the screen.

This message indicates that the PC is going through a series of built-in system checks. As each test is successfully completed a dot appears after the message.

If a test is not successful, a system error message will appear. See Part 2, Chapter 10, "Troubleshooting", for a list of system error messages, and what you should do.

If all is well, your PC will bleep and a message similar to the following will appear on the screen:

AMSTRAD PC 2086 640K (V1.1)
(c)1988 Amstrad plc

Please set time and date
Please fit new batteries

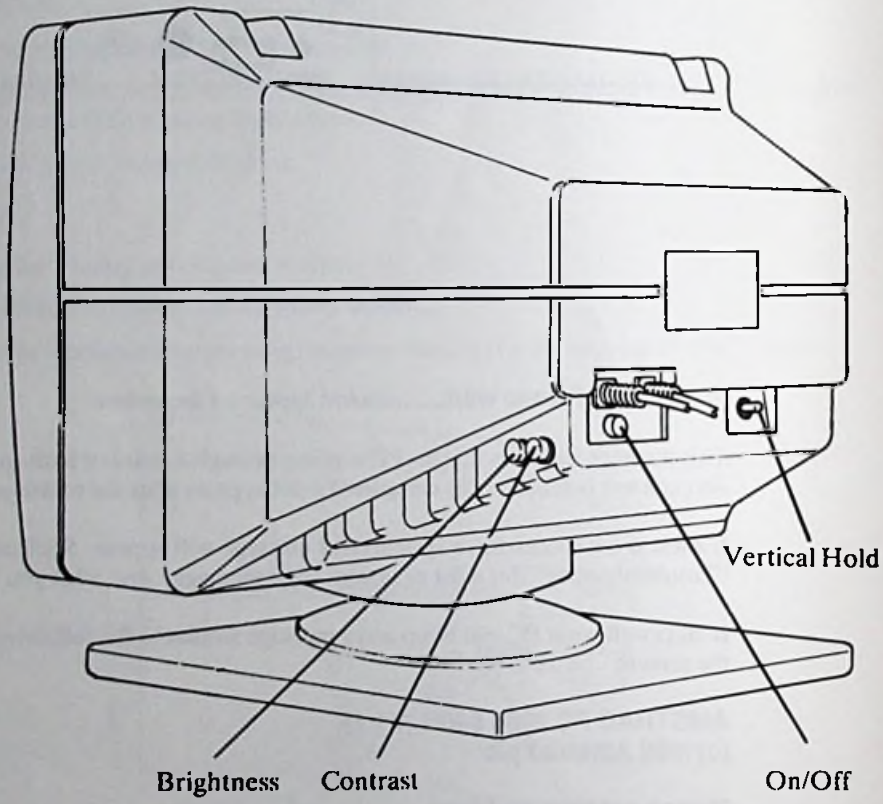
If you do not hear a 'bleep' or the sound is too loud, adjust the Volume Control on the left hand panel of the System Unit; below, and to the right of the Power switch.

If you have not connected the keyboard properly, or have not operated the keylock switch or something is holding down keys on the keyboard (or a mouse button), the following message will appear:

Check keylock switch, keyboard and mouse

Clear away anything resting on the keyboard and check that the keyboard is properly connected to the System Unit. Check that the connector on the side of the keyboard cable is firmly plugged into the socket on the side of the System Unit; check that the mouse is properly connected and that the mouse buttons are not held down by anything; and check that the keylock switch is unlocked. (The keylock is on the lefthand panel of the system unit). The message should disappear if you clear the problem.

Adjust the brightness and contrast of the messages on the screen to a comfortable level using the Brightness and Contrast controls on the right hand side of the Display.



If necessary, also adjust the V-Hold knob on the back of the Display until the display is steady.

The next step is to load the Operating System (MS-DOS). How you do this depends on whether you have a hard disk PC, a dual disk drive PC, or a single disk drive PC. Proceed with Step 7, "Load the Operating System", and follow the instructions appropriate to your particular PC (hard disk PC; dual disk drive PC; single disk drive PC).

Step 7. Load the Operating System (MS-DOS) - HARD DISK PC -

Check that the hard disk is prepared for use

Having switched on your PC, as described above, you should now see the drive letter, prompt and flashing cursor appear on the screen. This means your hard disk has already been configured and formatted for use.

If you see a message asking for the date or time just press the [↵] key for now.

MS-DOS (Microsoft Disk Operating System) is the operating system, and is loaded into memory, ready to be used. An explanation of these processes is included in Chapter 7, "The hard disk".

A description of MS-DOS is to be found in Chapter 2, "Guide to the PC for new users."

If you wish to reconfigure the hard disk (for example, to use more than one operating system) turn to Section 7.6 for instruction.

If the drive letter, prompt and flashing cursor have appeared on the screen, you can now go on to load the graphics interface software, Windows, which was supplied with your PC. Refer to your Windows User Guide for instructions on how to do this. If you decide to use Windows, remember to continue with the installation steps 8, 9 10 and 11.

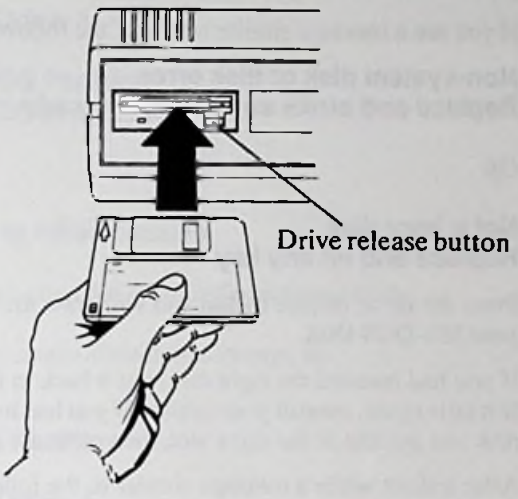
There is a remote possiblilty that your hard disk may not have been through the configuration and format processes. In which case the following message will appear on the screen:

Insert a SYSTEM disk into drive A
Then press any key

The hard disk must be configured and formatted before it can be used, therefore, proceed as follows:

WARNING:
This process will erase all information on your hard disk. If you think that your hard disk contains any useful programs or data please refer to your dealer to determine why MS-DOS was not loaded correctly as described above.

Select the MS-DOS Disk from your set of PC disks.



Hold the disk with arrow uppermost and pointing away from you.

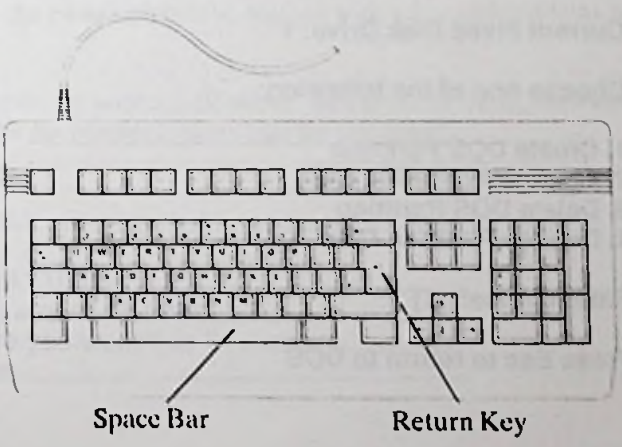
Gently insert the disk into the slot in drive A - the lefthand disk drive if you have two.

Push the disk all the way into the slot. The disk should simply slide in and latch into place.

To remove the disk from the drive, press the drive release button. The disk will slide a little way out of the slot. Grip the edge of the disk with your fingertips and remove it from the drive.

Practice inserting and releasing the disk a few times until you feel confident about doing this. Finish with the disk fully inserted into the drive.

Press either the space bar or the return key [↵] on the keyboard.



The PC then reads from this disk. You will see the green indicator light on the drive go on and off a few times while this is happening.

If you see a message similar to one of the following:

Non-system disk or disk error
Replace and strike any key when ready

OR

Not a boot disk
Replace and hit any key

Press the drive release button and withdraw the disk from the drive. Check that it is your MS-DOS Disk.

If you had inserted the right disk, put it back in the drive slot and press the space bar. If it fails again, consult your dealer. If you had inserted the wrong disk, select the right disk and put this in the drive slot, then press the space bar.

After a short while a message similar to the following will be displayed. If you see a message asking for the date or time just press the [↵] key for now.

MS-DOS Version 3.30
A>

† First of all, type in

FDISK

then press the [↵] key.

After a short while, you should see a message similar to the following

Fixed Disk Setup Program Version 3.30
(C)Copyright Microsoft Corp. 1987

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 Data

Enter choice: [1]

Press Esc to return to DOS

Press the [↵] key. The screen should change to:

Current Fixed Disk Drive: 1

1. Create Primary DOS Partition
2. Create Extended DOS Partition

Enter choice: [1]

Press Esc to return to Fdisk options

(A third option will be displayed if extended partitions exist)

Press the [↵] key. The screen should then change to:

Either A)

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

Or B)

Create Primary DOS Partition

Current Fixed Disk Drive: 1

Primary DOS Partition already exists

If the message on your screen is A) above then skip the next few paragraphs and continue with the paragraph below, marked with a ‡, which explains how to format the hard disk.

If the message on the screen is B) above, then press the [Esc] key twice to return to DOS. When the **A>** prompt appears type the command:

HDFORMAT

The screen will clear and the following message will be displayed:

DISK FORMAT UTILITY.....
This routine erases all data.....
Are you sure you want this ? .....
(Y)es or (N)o.....

Press **Y** to confirm that this is what you want to do. The message

Please wait.....

will then appear and after several minutes you will see the message:

Completed ! Returning to DOS.....

When the **A>** prompt appears return to the paragraph above marked with a † and continue on from there.

‡ Press the [↵] key. The screen should then change to:

System will now restart

Insert DOS diskette in drive A:
Press any key when ready...

Press the [↵] key. The computer will go through much the same procedure as when you first turned it on. (It is possible you may see some interference on the screen for a few seconds, this is perfectly normal and is not a fault with your PC) . Eventually, you will see an **A>** prompt.

Now type

SELECT A: C: 044 UK

The following message should appear:

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

Press [↵] to continue, you will then see the following message:

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

Type

Y and press [↵]

On the screen, you will see:

Head:0 Cylinder: 1

The numbers will change as you watch, and represent how far the computer has got as it prepares the hard disk. The cylinder number will give you some idea as to how far the process has reached. A 32 Megabyte disk will take about five minutes and count up to cylinder number 468.

When the computer has finished, the message

Format complete

will appear, followed shortly afterwards by something similar to

System transferred

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

If you don't want to give the newly formatted hard disk a label then just press the [↵] key. Otherwise type a name up to 11 characters long made up of the letters A..Z and digits 0..9 then press [↵]. The following message will then appear:

```
32602112 bytes total disk space
   79872 bytes used by system
32522240 bytes available on disk

Reading source file(s) . . .

  A:COMMAND.COM
  A:ANSI.SYS
  A:CONFIG.SYS
  A:COUNTRY.SYS
  A:DRIVER.SYS
  A:KEYBOARD.SYS
  A:PRINTER.SYS
  A:RAMDRIVE.SYS
  A:AUTOEXEC.BAT
  A:ASSIGN.COM
  A:BACKUP.COM
  A:CHKDSK.COM
  A:COMP.COM
  A:DEBUG.COM
  A:DISKCOMP.COM
  A:DISKCOPY.COM
  A:EDLIN.COM
  A:FDISK.COM
  A:FORMAT.COM
  A:GRAFTABL.COM
  A:GRAPHICS.COM
  A:KEYB.COM
```

```
A:LABEL.COM
A:MODE.COM
A:MORE.COM
A:PRINT.COM
A:RECOVER.COM
A:RESTORE.COM
A:SELECT.COM
A:SYS.COM
A:TREE.COM
A:DISPLAY.SYS
A:APPEND.EXE
A:ATTRIB.EXE
A:EXE2BIN.EXE
A:FASTOPEN.EXE
A:FC.EXE
A:FIND.EXE
A:JOIN.EXE
A:LINK.EXE
A:NLSFUNC.EXE
A:REPLACE.EXE
A:SHARE.EXE
A:SORT.EXE
A;SUBST.EXE
A:XCOPY.EXE
A:4201.CPI
A:5202.CPI
A:EGA.CPI
A:DATETEST.COM
A:DEVICE.COM
A:HDFORMAT.COM
A:PARK.COM
A:MOUSE.COM
A:RPED.EXE
A:VGATEST.EXE
        56 File(s) copied.
```

It is possible that the message will indicate a certain proportion of the disk as used in 'Bad Sectors'. This is perfectly normal.

The hard disk is now configured, formatted and has MS-DOS installed with the keyboard layout and date and time format of the country of your choice. (in this case the UK).

You should now have the drive prompt i.e. **A>** and the flashing cursor on the screen.

If you intend to use Windows, the presentation manager interface to MS-DOS, supplied with your PC, you can now install the contents of Windows Disks 1 - 5, on your hard disk. Refer to your Windows User Guide for instructions on how to do this.

Please note that Windows is provided on five diskettes, rather than the seven mentioned in the Windows User Guide, and that it is no longer recommended for use on a two drive floppy disk system.

Now continue with Step 8, "Set Time and Date"

Load the operating system - MS-DOS

Your PC was supplied with six floppy disks:

Microsoft® MS-DOS Disk

Disk 1/5 - Microsoft® Windows Setup Disk

Disk 2/5 - Microsoft® Windows Build Disk

Disk 3/5 - Microsoft® Windows Utilities Disk

Disk 4/5 - Microsoft® Windows Fonts Disk

Disk 5/5 - Microsoft® Windows Desktop Applications and Microsoft® Windows Write Disk

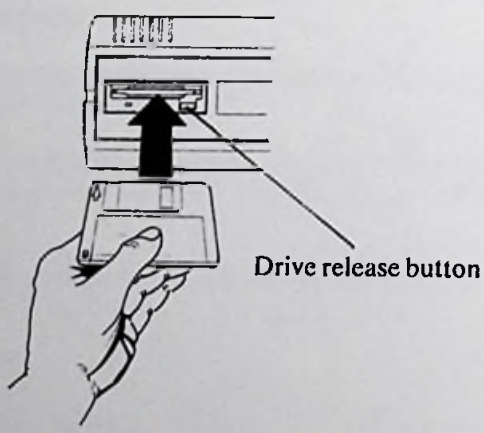
You must now go through a short process to load MS-DOS, installed on the MS-DOS disk, into your PC's memory. You will need to do this every time you switch on your PC and want to use MS-DOS.

See Chapter 2, "Guide to using the PC for new users", and Part 2, "Guide to using MS-DOS", for information about MS-DOS.

Having switched on as described in step 6, "The startup procedure", you should now have the following message displayed on your screen

Insert a system disk into drive A
Then press any key

Select the MS-DOS Disk from your set of PC disks.



Hold the disk with the arrow uppermost and pointing away from you.

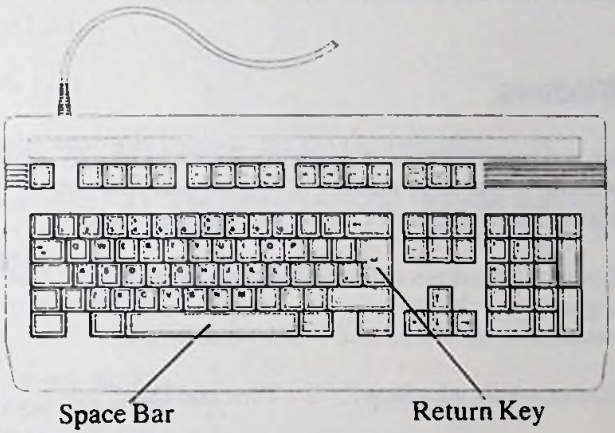
Gently insert the disk into the slot in drive A- the lefthand disk drive.

Push the disk all the way into the slot. The disk should simply slide in and latch into place.

To remove the disk from the drive, press the drive release button. The disk will slide a little way out of the slot. Grip the edge of the disk with your fingertips and remove it from the drive.

Practice inserting and releasing the disk a few times until you feel confident about doing this. Finish with the disk fully inserted into the drive.

Press either the space bar or the return key [↵] on the keyboard.



The PC then reads from this disk. You will see the drive indicator light go on and off a few times while this is happening.

If you see a message similar to one of the following:

Non-System disk or disk error
Replace and strike any key when ready

OR

Not a boot disk
Replace and hit any key

Release the disk from the drive by pressing the release button. Withdraw the disk from the drive. Check that it is the MS-DOS Disk.

If you had inserted the right disk, put it back in the drive and press the space bar. If it fails again, consult your dealer. If you had inserted the wrong disk, replace this disk in its paper cover and take the right disk out of its cover. Put this disk in the drive. Then press the Space Bar.

After a short while, a message similar to the following will be displayed. If you see a message asking for the date or time just press the [↵] key for now.

MS-DOS Version 3.30
A>

This is the MS-DOS command line. If you want to, you can start using the MS-DOS operating system right away. The first thing you can do is set the time of your PC's clock . See step 8, below, "Set time and date".

Windows

It is not recommended that you use Windows on a PC which does not have a hard disk because of the large number of system files that you would have to load, every time you switched on your PC and wanted to use Windows.

However, Windows software has been supplied with your PC to use if, or when, you upgrade your PC to include a hard disk. Refer to your Windows User Guide for instructions on how to load Windows software.

Now continue with Step 8, "Set Time and Date"

Step 8. Set Time and Date

At this stage it is a good idea to set the time of your PC's clock. This is a 24 hour clock maintained in battery-backed RAM (memory). You can do this by using the MS-DOS TIME command, as follows:

At the **A>** or **C>** prompt, type **TIME** and press the Return [↵] key. The current setting is displayed and then you are asked to type in a new one. If the time displayed is correct, just press the Return key [↵]. If not, enter the new time as: **hh:mm**

where:

hh represents the hour (00...23)
mm represents the minute (00...59)

For example to set the time to 4.00pm (16.00 hours) type

TIME and press [↵]

16:00 and press [↵]

Set the date in the same way by using the DATE command.

Type **DATE** and press [↵]. The current setting is displayed and then you are asked to type in a new one. If the date displayed is correct, just press [↵]. If not, enter the new date as: **dd-mm-yy**

where:

dd represents the day (0...31)
mm represents the month (01...12)
yy represents the last two digits of the year (00...99)

For example, to set the date to the 26th of December 1988, type:

DATE and press [↵]

26-12-88 and press [↵]

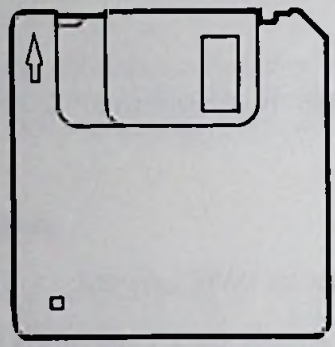
Note that the TIME and DATE commands are explained in detail in Part 2, Chapter 8.

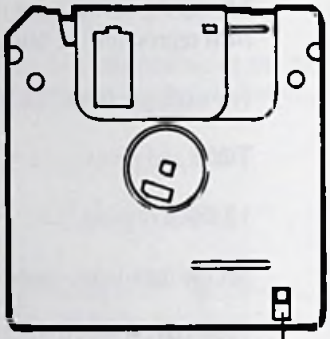
Step 9. Back-up your PC Disks

Having gone through the startup procedure (Step 6), and loaded MS-DOS (Step 7), you should now produce duplicate copies of the your PC's MS-DOS disk.

The reason for doing this is to ensure that you can still use your PC even if the disk becomes damaged or you accidentally erase all the contents.

First of all, make sure that the write-protect shutter is **OPEN** on the MS-DOS disk, that is, the disk you are going to copy, so that the disk is protected against possible accidents while copying.





If you have a single-drive PC

At the **A>** prompt type:

DISKCOPY A: A:

and then press the [↵] key

After a short while you will see the following message on the screen:

Insert SOURCE diskette in drive A:

Press any key when ready . . .

The SOURCE diskette is the disk that holds the data you want to copy. The disk you are going to copy is the MS-DOS Disk.

Hold the disk with the arrow uppermost and pointing away from you. Insert the disk all the way into the drive until it latches into place. Then press the [↵] key. You should see the light on the drive come on for a while and then the message:

Copying 80 tracks
9 Sectors/Track, 2 Side(s)

Followed by

Insert TARGET diskette in drive A:

Press any key when ready . . .

The Target disk is the disk you want to store the copy on, that is, a new blank disk.

Release the disk in the drive by pressing the release button on the drive, and withdraw the disk you are copying.

Take hold of a new blank disk with the arrow uppermost and pointing away from you. Slide it into the slot in the drive until it latches into place. Then press the [↵] key.

You will then see the following message on the screen:

Formatting while copying

This is perfectly normal. It just tells you that your PC is marking out your disk into storage segments at the same time as making the copy.

The process of inserting the source disk and then the target disk will continue until all of the disk has been copied.

Finally you will see the message:

Copy another diskette (Y/N)?

Then type **N** in response to this question.

Press the release button, and remove the disk from the drive. Write "MS-DOS Disk" on a blank disk label, and stick it on the disk.

You will then have two PC Disks. Keep one as a Master disk, which you use only to make further copies if your second disk becomes damaged. Use the second disk whenever you want to load MS-DOS.

When you have finished the copying process, make sure that the write protect shutter is OPEN on both disks. This will prevent any data being written onto either disk, by accident.

If you have a dual drive PC

At the **A>** prompt type:

DISKCOPY A: B:

and press the [↵] key.

After a short while, you will see the following message on the screen.

Insert SOURCE diskette in drive A:

Insert TARGET diskette in drive B:

Press any key when ready . . .

The SOURCE diskette is the disk that holds the data you want to copy. The source disk you are going to copy is the MS-DOS Disk. Select this disk and hold it with the arrow uppermost and pointing away from you. Insert the disk all the way into drive A, the lefthand drive, until it latches into place.

The TARGET diskette is the disk you want to store the copy on. That is, a blank floppy disk. Select a blank disk and hold it with the arrow uppermost and pointing away from you. Insert the disk all the way into drive B, the righthand drive, until it latches into place.

Then press the [↵] key.

After a short pause, you will see the message:

Copying 80 tracks
9 Sectors/Track, 2 Side(s)

Because your target disk is new, you will see the following message on the screen:

Formatting while copying

This is perfectly normal. It just tells you that your PC is marking out your disk into storage segments at the same time as making the copy.

Finally, you will see the message:

Copy another diskette (Y/N)?

Type **N** in response to this question.

Remove the disk from drive B by pressing the drive release button. Write "MS-DOS Disk" on a blank disk label and then stick it on the disk, next to the manufacturers label.

Then remove the SOURCE disk from drive A.

You will then have two MS-DOS Disks. Keep one as a Master disk, which you use only to make further copies if your second disk becomes damaged. Use the second disk whenever you want to load MS-DOS.

When you have finished the copying process make sure that the write protect shutter is **OPEN** on both disks. This will prevent any data being written onto either disk, by accident.

If you have a hard disk PC

If you have a hard disk PC, it should not be necessary to make a duplicate copy of any of the six PC floppy disks, as they have already been copied onto your hard disk. If you do want another copy for security, you can use the copying instructions given for the single drive PC, above, which you should repeat five times to copy all six disks.

For instructions on how to make back up copies of the hard disk see the BACKUP command in Part 2, Chapter 8.

Step 10. How to Reset your PC

Resetting your PC clears its memory so that you can start using the machine afresh. It is often used to change from using one Application software package to another, or after a program has failed.

Do not reset your PC before you have checked whether there is any information, currently held in its memory, that you want to save to disk.

Resetting your PC clears its memory. Any information that has not been saved to disk will be lost.

The steps are as follows:

1. Press the drive release button(s) and withdraw the disk(s) from the disk drive(s).
2. Hold down both the **[Ctrl]** and **[Alt]** keys and press the **[Del]** key.
3. If you have a floppy disk drive PC insert your Startup disk (MS-DOS disk) into drive A (your lefthand disk drive if you have two). Then press the [↵] key.

Note: In the case of the Hard Disk PC there is no need to insert a Startup (MS-DOS) disk, unless you want to override the operating system loaded automatically from hard disk.

Step 11. Switching off your PC

1. Check that your PC isn't reading any data from your disks or writing data to them.
2. Press the disk drive release button(s). This automatically releases the disks from the drive.
3. Withdraw the disk(s) from the drive(s).
4. Only then, switch off your PC by pressing and releasing the power button:

ALWAYS REMEMBER TO REMOVE YOUR DISK(S) FROM THE DISK DRIVE(S) BEFORE YOU SWITCH OFF.

ALWAYS WAIT FOR AT LEAST FIVE SECONDS BEFORE SWITCHING BACK ON.

3.2 The PC system disks

The contents of the six disks are as follows:

MS-DOS disk

Filename		Size in bytes
COMMAND	COM	25276
ANSI	SYS	1647
CONFIG	SYS	40
COUNTRY	SYS	11254
DRIVER	SYS	1165
KEYBOARD	SYS	19735
PRINTER	SYS	13559
RAMDRIVE	SYS	6481
AUTOEXEC	BAT	103
ASSIGN	COM	1530
BACKUP	COM	29976
CHKDSK	COM	9819
COMP	COM	4183
DEBUG	COM	15866
DISKCOMP	COM	5848
DISKCOPY	COM	6264
EDLIN	COM	7495
FDISK	COM	48919
FORMAT	COM	11671
GRAFTABL	COM	6136
GRAPHICS	COM	13943
KEYB	COM	9041
LABEL	COM	2346
MODE	COM	15440
MORE	COM	282
PRINT	COM	8995
RECOVER	COM	4268
RESTORE	COM	35650
SELECT	COM	4132
SYS	COM	4725
TREE	COM	3540
DISPLAY	SYS	11259
APPEND	EXE	5794
ATTRIB	EXE	10656
EXE2BIN	EXE	3050
FASTOPEN	EXE	3888
FC	EXE	15974

Filename		Size in bytes
FIND	EXE	6403
JOIN	EXE	9612
LINK	EXE	39172
NLSFUNC	EXE	3029
REPLACE	EXE	13234
SHARE	EXE	8608
SORT	EXE	1946
SUBST	EXE	10552
XCOPY	EXE	11216
4201	CPI	17089
5202	CPI	459
EGA	CPI	49065
DATETEST	COM	17
DEVICE	COM	4838
HDFORMAT	COM	2167
PARK	COM	887
MOUSE	COM	8669
RPED	EXE	4644
VGATEST	EXE	26158

Windows Setup Disk (1/5)

Filename		Size in bytes
SETUP	EXE	44039
SETUP	INF	21704
KERNEL	EXE	57088
MSDOS	EXE	46224
MSDOSD	EXE	6490
SPOOLER	EXE	14640
USER	EXE	154688
WIN	CNF	3952
WIN	INI	2176
COMM	DRV	4396
SOUND	DRV	5309
SYSTEM	DRV	2725
GDI	EXE	101706
WINOLDAP	MOD	60112
GWBASIC	EXE	80592

Windows Build Disk (2/5)

Filename		Size in bytes
EGAHIBW	DRV	26848
EGAHIRES	DRV	33040
EGAHIRES	GRB	3573
EGALORES	DRV	30208
EGAMONO	DRV	26960
EGAMONO	GRB	3358
EGAMONO	LGO	446
HERCULES	DRV	28272
HERCULES	GRB	2615
HERCULES	LGO	557
IBMPS230	DRV	26496
IBMPS250	DRV	32720
CGA	DRV	24784
CGA	GRB	2672
CGA	LGO	468
FONTHI	FON	4608
FONTHIUS	FON	4528
FONTLO	FON	4160
FONTLOUS	FON	4528
FONTSQ	FON	5728
FONTSQUS	FON	5536
KBDFR	DRV	6145
KBDFS	DRV	6113
KBDGR	DRV	6129
KBDIT	DRV	5873
KBDNE	DRV	6177
KBDSP	DRV	6177
KBDUK	DRV	5841
LMOUSE	DRV	3298
MOUSE	DRV	3667
MSMOUSE1	DRV	2014
MSMOUSE2	DRV	2014
NOMOUSE	DRV	1245
USA	DRV	3836
MSLOGO	LGD	2570
MOUSE2	DRV	4017
KBDUS	DRV	5585
KBDCA	DRV	6081
KBDBE	DRV	6129
KBDSG	DRV	6185
KBDSF	DRV	6129
KBDLA	DRV	5953
KBDPO	DRV	6081

Filename		Size in bytes
KBDDA	DRV	6081
KBDNO	DRV	6113

Windows Utilities Disk (3/5)

Filename		Size in bytes
CITOH	DRV	14208
EPSON	DRV	17168
EPSONMX	DRV	14240
HPPCL	DRV	100592
HPPLOT	DRV	62560
IBMCOLOR	DRV	14144
IBMGRX	DRV	14512
EPSONLQ2	DRV	47616
NECP2	DRV	14352
PROPRINT	DRV	32944
TTY	DRV	6224
READMEEP	TXT	1060
READMEHP	TXT	9550
READMEPL	TXT	1087
READMEPR	TXT	975
EGA	SYS	2984
EMM	SYS	54528
EMM	TXT	13478
HARDWARE	TXT	565
PS2EMM	SYS	13008
PS2EMM	TXT	8770
PSCRIPT	DRV	121184
PSPREP	TXT	18841
RAMDRIVE	SYS	8225
RAMDRIVE	TXT	7776
READMEPS	TXT	9196
REMM	SYS	15643
REMM	TXT	1770
SMARTDRV	SYS	10082
SOFTWARE	TXT	409
THINKJET	DRV	13296
TI850	DRV	12816
TOSH	DRV	14480
XER4020	DRV	22032

Windows Fonts Disk (4/5)

Filename		Size in bytes
NEWFON	EXE	33231
PCLPFM	EXE	24966
MODERN	FON	7584
ROMAN	FON	11120
SCRIPT	FON	10304
COURB	FON	19088
HELVB	FON	50880
TMSRB	FON	45936
COURE	FON	23808
HELVE	FON	64784
TMSRE	FON	58304
COURA	FON	14144
HELVA	FON	36768
TMSRA	FON	35392
COURC	FON	13040
HELVC	FON	38960
TMSRC	FON	37824
COURD	FON	21328
HELVD	FON	58144
TMSRD	FON	57184

Windows Desktop Applications and Microsoft Write Disk (5/5)

Filename		Size in bytes
CALC	EXE	28000
CALENDAR	EXE	38896
CARDFILE	EXE	39264
CLOCK	EXE	8960
CONTROL	EXE	57216
CVTPAINT	EXE	5712
NOTEPAD	EXE	19072
PAINT	EXE	93280
REVERSI	EXE	15552
TERMINAL	EXE	48640
ABC	TXT	42
DOTHIS	TXT	493
CLIPBRD	EXE	10800
README	TXT	15199
WRITE	EXE	198336
PRACTICE	WRI	2944
PIFEDIT	EXE	30672

3.3 System setup using the DEVICE command

Introduction

There is a small amount of memory within your PC known as the Non Volatile RAM which stores information on how the system should be set-up when the PC is switched on. When the PC is switched off this data would be lost if the memory were not powered by the batteries within the computer. It is therefore, sometimes known as the battery-backed RAM.

This memory resides within the clock chip and some of the storage area is used to hold the current date and time. However, details of the floppy disk drives are also stored there.

The details of the floppy disk drives may be changed using the DEVICE command. The DEVICE command may also be used to affect the operation of the display adaptor.

DEVICE

Use the DEVICE command to display or modify the PC video or floppy disk configuration parameters.

- **Configuration Display**

The DEVICE command will display reports on the following:

- the serial and parallel ports
- the video parameters
- disk drives
- attached standard units, for example, an adaptor port or a data migration adaptor

Form **DEVICE**

Example

If you want to see reports on the current system configuration parameter settings, type:

DEVICE

The screen will display a report which looks something like this:

```
Amstrad PC2086 Device Assignments (1.0)
Copyright (C) Amstrad plc 1988

SERIAL/PARALLEL PORTS
COM1: 03F8 (Internal)          LPT1: 0378 (Internal)
COM2: Not Assigned             LPT2: Not Assigned
                               LPT3: Not Assigned
VIDEO
Display Adaptor..... Internal PVGA (Colour-80)
Current Video Mode..... 03h
Multi-Frequency Mode..... Disabled
Auto Mono/Colour switching.. Enabled
Display Option Switch 3..... Off
Display Option Switch 4..... On

DISK DRIVES AND ADAPTORS
Drive A type is 720K Bytes, 3.5 inch, With Changeline
Drive B type is 360K Bytes, 5.25 inch, Without Changeline
Maths Co-processor (NDP) Not Fitted
Hard Disk Drive C: 468 tracks, 8 heads, 17 sectors/track (32Mb)
```

The above report indicates:

- The serial and parallel ports consist of the standard on-board peripherals.

(additional serial and parallel cards will be reported along with the I/O addresses to which the device is assigned)

- A color display is attached and that the system is displaying in color text mode (mode 03h).

- Multi-frequency mode (DIP switch 5) is switched OFF. This is the setting for fixed frequency displays, and is the default setting for your PC. To change the setting to Multi-frequency display, switch the number 5 DIP switch to ON. The DIP switches are to be found at the back of the system unit.

Display Selector
DIP Switches

- The Auto Mono/Colour switch (DIP switch 6) is set to ON. This is the default setting for your PC. This switch selects the manner in which the VGA card controls switching between colour and monochrome modes, that is, if you have a monochrome display, you can use software to switch to colour mode, and if you have a colour display you can switch to monochrome mode. (See Chapter 4 for more details).

You should only change this switch to OFF if you have a specific reason to restrict this facility.

- The standard 720K floppy is installed in the drive A position and that, most likely, there is an external 360K floppy disk attached to the B drive.

- The hard disk (Drive C:) is a fairly typical 32Mb unit.

- Some industry standard units are attached. For example, an 8087 Maths Co-processor, or a games adaptor port, or a data migration adaptor fitted to the parallel (bi-directional) centronics port.

Setting Configuration with the DEVICE command

The DEVICE command allows you to set the internal PVGA (Paradise Video Graphics Array) adaptor to a number of optional modes or to set the floppy disk BIOS interface parameters.

Form DEVICE [SETFD|DISPLAY]

If you use the DEVICE command with a parameter it does not understand, such as **DEVICE ?**, you will be prompted to specify the top level parameters for display or floppy disk set up, as follows:

Usage: DEVICE [SETFD|DISPLAY]

This is to tell you that you must use either the parameter SETFD, or the parameter DISPLAY, in order to select one of the two options.

Setting the floppy disk parameters

When you modify the floppy disk hardware configuration, you will have to tell the PC about the new configuration.

To do this, use the following form of the DEVICE command:

Form DEVICE SETFD *[/D:n][/T:n][+C/-C][+S/-S][+P/-P]*

where:

/D:*n* specifies drive selection (where *n* is 0 or 1 representing the default physical drives A and B)

/T:*n* specifies the drive type (where *n* is 0, 1 or 2 representing types as follows:

0 - 360Kb 5.25 floppy disk drive without changeline

1 - 720Kb 3.5 inch floppy disk drive with changeline

2 - 1.44Mb 3.5 inch floppy disk drive with changeline

+C Turn on changeline in order to override the defaults above

-C Turn off changeline in order to override the defaults above

+S Turn on drive a-b swapping

-S Turn off drive a-b swapping

+P Turn on write precompensation

-P Turn off write precompensation

Drive A-B swapping

Drive A-B swapping is a special ROS function built into the PC which enables remapping all BIOS requests for drive A to go to drive B and vice-versa.

This is useful when a 360K 5.25 inch floppy adaptor is fitted, and the software must run from the A drive. This option works with about 90 per cent of the protected software with notable exceptions, such as LOTUS 1-2-3, which bypasses the ROM BIOS and goes directly to the FDC (Floppy Disk Controller) hardware to check that its key diskette is installed.

Write precompensation

Write precompensation defaults to ON and is normally required by all drives that are in standard industry usage.

Note that this parameter sets the FDC hardware and effects both drives.

All SETFD parameters are written to the NVR (Non-Volatile RAM in the RTC) and will not be passed through to the operating system, until the following soft reset (hold down [Ctrl] and [Alt] and press [Del]) or the next time the machine is switched on. Write Precompensation is an exception to this rule and will take effect immediately.

If you type **DEVICE SETFD** with no parameters, or parameters which it does not understand, you will receive a prompt similar to the following:

Usage: DEVICE SETFD [/D:n /T:n [+C|-C]] [+S|-S] [+P|-P]

If you type an invalid parameter, such as **DEVICE SETFD /D:3 /T:O** you will receive a response like the following:

Invalid parameter

Setting the PC2086 PVGA options

Use the following form to set the internal PVGA options:

Form DEVICE DISPLAY *mode* **[/B]**

where:

mode selects one of the following:

CGA	to set colour graphics adaptor emulation mode
MDA	to set monochrome adaptor emulation mode
HERCO	to set Hercules half graphics emulation mode
HERC1	to set Hercules full graphics emulation mode
VGA	to return to full VGA adaptor mode

/B optional parameter to bootstrap the machine immediately after setting the display option

Note The **/B** parameter is used when bootstrapping into a protected environment which assumes a display adaptor, such as a CGA, is present.

When you enter a parameter that DEVICE does not understand as a DISPLAY option, or you enter **DEVICE DISPLAY** with no parameters you will receive a prompt like the following:

Usage: DEVICE DISPLAY [CGA|MDA|HERCO|HERC1|VGA [/B]]

You can use the form, DISPLAY, to override the default VGA colour/monochrome established, because the system sensed a colour or a monochrome monitor attached at system startup.

Example

- If you have a monochrome monitor but want to view colour VGA software (as grey scales) on your monitor, use two successive DEVICE commands, as follows:

  DEVICE DISPLAY CGA

  DEVICE DISPLAY VGA

  The first command forces colour mode and the second command re-establishes the VGA mode.

- If you have a colour VGA monitor and want to obtain a monochrome configuration use the following two command lines:

  DEVICE DISPLAY MDA

  DEVICE DISPLAY VGA

Note It is recommended, for safety, to use the **/B** parameter when a totally different primary display configuration is established:

That is, when changing from monochrome to colour, or from colour to monochrome.

If this precaution is not taken, MS-DOS may become confused when dealing with the display where options, such as code page switching, have been enabled: even seemingly normal display operations may produce unexpected results.

3.4 CONFIG.SYS file

As MS-DOS loads from disk into your PC it examines a text file on the startup disk called CONFIG.SYS. There are nine special commands, explained below, that can appear in this file. These commands instruct MS-DOS to change certain default options for things like the letter used for the last drive or how much space it should reserve for buffering information from disk.

Command	Description
BREAK	Extended BREAK checking [Ctrl][C]
BUFFER	Number of sector buffers.
COUNTRY	Country Specific parameter selection
DEVICE	Device driver installations.
DRIVPARM	Override the drive parameters for a logical drive.
FCBS	Maximum number of file handles open concurrently.
LASTDRIVE	Maximum drive letter allowable.
SHELL	Top level command processor specification.
STACKS	Override the default DOS stack resources.

The CONFIG.SYS can be created with any text editor and the simple screen editor RPED is ideal for this purpose.

3.4.1 BREAK Command

This command enables the MS-DOS extended break checking to be either set or reset. Normally, MS-DOS checks to see if [Ctrl] [C] has been typed while it is reading from the keyboard, writing to the screen or a printer. Setting Break to **ON** allows [Ctrl][C] checking to be extended to other functions such as disk reads and writes.

The syntax of the BREAK command is:

BREAK=[ON]

or

BREAK =[OFF]

If no field is specified then **OFF** is assumed (as the default value).

3.4.2 BUFFERS Command

This command allows you to specify the number of buffers that MS-DOS allocates when it starts up. A disk buffer is a block of memory where MS-DOS holds data being read from, or written to, a disk when the amount of data is not an exact multiple of sector size.

The syntax of the BUFFERS command is:

BUFFERS=*n*

Where *n* is a number between 2 and 255. If the BUFFERS command is not used then MS-DOS defaults to 2 buffers. The number of buffers remains in effect after MS-DOS is loaded until the machine is switched off or MS-DOS is loaded again.

For best performance for standard applications (word processors, spreadsheets, etc.) a buffers allocation between 10 and 20 is recommended. If you tend to use many subdirectories then an allocation upwards to 30 may be better. But since buffers use the system available memory, there may have to be a compromise between memory usage and performance. Buffers allocated beyond 40 serves no useful purpose. Refer to the User Manuals for your applications, if in doubt about required buffers for particular applications programs.

3.4.3 COUNTRY Command

The country command is used to select the country dependent information as shown in the following table.

The syntax of the COUNTRY command is:

COUNTRY=*<country code> [,<code page>] [,[drive:]<filename>]*

<country code>	is shown in the **Num** column in the following table.
<code page>	is the code page for the country. Shown in the **DcP** column in the following table (see also chapter 5).
<filename>	is a file containing the country information. If no filename is specified the file COUNTRY.SYS is assumed and it must be available in the root directory.

Note that the utility, NLSFUNC, can also be used to load country specific information.

Country	Num	KbC	DcP	AcP	DtF	DtS	TmS	TmF	CSm	CFt	CSd	ThS	DeS	DlS
Australia	061	US	437	850	1	-	:	0	$	0	2	,	.	,
Belgium	032	BE	437	850	1	/	:	1	BEF	2	2	.	,	;
Canada (Eng)	001	US	437	850	0	-	:	0	$	0	2	,	.	,
Canada (Fr.)	002	CF	863	850	2	-	:	1	$	3	2	,	,	;
Denmark	045	DK	865	850	1	-	.	1	kr	2	2	.	,	;
Finland	358	SU	437	850	1	.	.	1	mk	3	2	,	,	;
France	033	FR	437	850	1	/	:	1	F	3	2	,	,	;
Germany	049	GR	437	850	1	.	.	1	DM	2	2	.	,	;
Italy	039	IT	437	850	1	/	:	1	L	0	0	.	,	;
Israel	972	--	437	850	1	.	:	1	ō	2	2	.	.	,
Latin America	003	LA	437	850	1	/	:	1	$	3	2	,	.	;
Middle East	785	--	437	850	1	/	:	0	ñ	3	3	.	,	;
Netherlands	031	NL	437	850	1	-	:	1	f	2	2	.	,	;
Norway	047	NO	865	850	1	/	.	1	Kr	2	2	.	,	;
Portugal	351	PO	860	850	1	/	:	1	$	4	2	.	,	;
Spain	034	SP	437	850	1	/	:	1	Pt	3	2	.	,	;
Sweden	046	SV	437	850	2	-	.	1	SEK	2	2	.	,	;
Switzerland(Fr.)	041	SF	437	850	1	.	.	1	Fr	2	2	'	.	,
Switzerland(Ger)	041	SG	437	850	1	.	.	1	Fr	2	2	'	.	,
United Kingdom	044	UK	437	850	1	-	:	1	£	0	2	,	.	,
United States	001	US	437	850	0	-	:	0	$	0	2	,	.	,

Where:

Num	=	COUNTRY Number Code
KbC	=	KEYB code
DcP	=	Default Code Page
AcP	=	Alternate Code Page
DtF	=	Date format (0=US M/D/Y, 1=Euro D/M/Y, 2=Japan Y/M/D)
DtS	=	Date Separator
TmS	=	Time Separator
TmF	=	Time Format (0=12 hour clock, 1=24 hour clock)
CSm	=	Currency symbol
CFt	=	Currency Format (Bit 0=0 currency sumbol precedes, 1=follows field Bits 1&2 = number of spaces between value and symbol)
CSd	=	Number of significant digits in currency
ThS	=	Thousands Separator
DeS	=	Decimal Separator
DlS	=	Data List Separator

The items in this table are used in conjunction with the COUNTRY command and the Keyboard Utility (KEYB).

3.4.4 DEVICE Command

This command installs the device driver in the specified pathname to the system list.

The syntax of the DEVICE command is:

DEVICE=*[<drive:>]<pathname>*

The file specified is loaded and given control. The driver may then perform the necessary steps to configure itself and the sysytem for its operation. See the MS-DOS Technical Reference Manual for information on how to create your own device driver.

Your MS-DOS disk contains six installable device drivers, DRIVER.SYS, RAMDRIVE.SYS, DISPLAY.SYS ANSI.SYS, KEYBOARD.SYS, and PRINTER.SYS, which can be used for variable device configurations.

If you plan to use the ANSI escape sequences described in the technical information at the end of Chapter 4, you would include the following command in your CONFIG.SYS file:

DEVICE=ANSI.SYS

This command causes MS-DOS to replace all keyboard input and screen output support with the ANSI escape sequences. Refer to Chapter 4 for ANSI escape sequence reference information.

3.4.4.1 DRIVER.SYS

DRIVER.SYS is an installable device driver that supports external drives. To install DRIVER.SYS, include the following command in your CONFIG.SYS file:

DEVICE=DRIVER.SYS/D:*dd***[/C][/F:***ff***][/H:***hh***][/N][/S:***ss***][/T:***tt***]**

Where:

/D:*dd*	is drive number (0-127:Floppy drive, 128-255 Hard drives)
and optionally:	
/C	indicates changeline (doorlock) support required.
/F:*ff*	indicates the form factor where:

0 = 5.25 inch floppy diskette, 320/360 K bytes.
1 = 5.25 inch floppy diskette, 1.2 M bytes
2 = 3.5 inch floppy diskette, 720 K bytes.
3 = 8 inch floppy diskette, Single Density.
4 = 8 inch floppy diskette, Double Density.
5 = Hard Disk.
6 = Tape Drive.
7 = 3.5 inch floppy diskkette, 1.44 M bytes.

/H:*hh*	is the maximum head number(1-99)
/N	indicates non-removable block device.
/S:*ss*	is the number of sectors per track(1-99).
/T:*tt*	is the number of tracks per side(1-999).

3.4.4.2 RAMDRIVE.SYS

RAMDRIVE.SYS is an installable device driver which enables the usage of a portion of the computer's memory as though it were a disk drive. This area of memory is referred to as a RAM disk or a virtual disk.

If you have an expanded memory card which meets the LIM [Lotus®/Intel®/Microsoft®] Expanded Memory Specification, you can use this memory for one or more RAM disks. Otherwise RAMDRIVE.SYS locates RAM drives in system memory.

To install RAMDRIVE.SYS, include the following command in your CONFIG.SYS file.

DEVICE=RAMDRIVE.SYS *[<disk size> [<sector size> [<entries>]]] [***/A***]*

Where:

<disk size>	is disk size in kilobytes. Default is 64 and minimum is 16.
<sector size>	is the sector size. The value 128, 256, 512, and 1024 are allowed. Default 128.
<entries>	is the number of root directory entries. The default is 64, the minimum value is 4 and the maximum is 1024.

/A indicates that a memory board which meets the LIM Expanded Memory Specification for a RAM drive is in use. If this switch is used, the **/E** switch cannot be used.

There is an additional parameter for this driver which applies to 80286 style CPU architecture with memory above the 1 M byte range. This parameter is as follows:

/E indicates that extended memory (above 1MB) is to be used. If this switch is used, the **/A** switch cannot be used. Extended memory cannot be fitted to an 8086 based PC.

3.4.4.3 DISPLAY.SYS

DISPLAY.SYS is an installable device driver which supports code page switching for the console device. A code page is an alternative set of 256 characters which can be used in graphic display modes and with EGA display adapters in text modes.

To install DISPLAY.SYS insert a command line of the following form in your CONFIG.SYS file:

DEVICE=*[<drv:>][<path>]***DISPLAY.SYS CON***[:]=[<type>[,<hwcp>[,n,m]]*

The **CON=** parameters are as follows:

<type>	The display adapter in use consisting of MONO,CGA,EGA or LCD.
<hwcp>	The code page supported by the hardware as follows: 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. For EGA, n can be 2.
m	The number of sub-fonts that are supported by each code page.

Files EGA.CPI and LCD.CPI are additional code page files for use with the MODE command of the form:

MODE con: cp prepare=((*<cplist>***)** *[drive:]* *<cpfile>***)**

where:

<cplist>	Is 850 if the hardware code page (hwcp) is 437. If the hardware code page is not code page 437, cplist is 850 plus the hardware code page. For example, Portuguese is (850,860).
drive:	specifies the drive path where the display code page font file is loaded.
<cpfile>	is the name of the code page font file.

Note that this MODE commmand should be used in the AUTOEXEC.BAT file.

3.4.5 DRIVPARM command

The DRIVPARM command allows overriding of the device parameters for a specific logical drive.

The syntax is:

DRIVPARM=/D:*dd* **[/F:***ff* **/T:***tt* **/S:***ss* **/N /C /H:***hh***]**

Where:

/D:*dd*	is drive number (0-255) (0=A, 1=B, 2=C...)
and optionally:	
/T:*tt*	is the number of tracks per side (1-999).
/S:*ss*	is the number of sectors per track (1-99).
/H:*hh*	is the maximum head number (1-99).
/C	indicates changeline (doorlock) support required.
/N	indicates non-removable block device.
/F:*ff*	indicates the form factor where:

0=5.25 inch floppy diskette, 320/360K bytes.
1=5.25 inch floppy diskette, 1.2M bytes.
2=3.5 inch floppy diskette, 720K bytes.
3=8 inch floppy diskette, Single density.
4=8 inch floppy diskette, Double density.
5=Hard drive.
6=Tape Drive.
7=3.5 inch floppy diskette, 1.44M bytes.

This command allows the overriding of default system parameters for a particular logical drive. This information would be used by the commands which create new diskettes (such as FORMAT and DISKCOPY) when writing out the directory and FAT (File Allocation Table) information. The information in the FAT is used when determining device characteristics for floppy disks, hard disks and tape drives.

If no form factor (**/F:***ff*) is specified then a value of 2 is assumed (720K, 3.5 inch diskette).

3.4.6 FCBS command

The FCBS command allows you to specify the number of file control blocks available to the system and consequently the number of files which can be opened at any one time.

The syntax of the FCBS command is:

FCBS=*x* , *y*

Where *x* is the number of FCBs (in the range 1 to 255) to allocate, and *y* is the number of FCBs protected from automatic closure when a program tries to open more than *x* files. The first *y* files opened will be protected. MS-DOS selects the least recently used (non-protected) FCB when it must automatically close a file.

If the FCBS command is not used MS-DOS defaults *x* and *y* to 4 and 0 respectively. It is an error to set *y* greater than *x*.

3.4.7 FILES command

The FILES command specifies the maximum number of file handles that can concurrently be opened. When a program opens a file or a device it is assigned an identifier or "handle" which can be used by that program in referring to the file.

The syntax of the FILES command is:

FILES=*n*

where *n* is the number of handles in the range 8 to 255. When no FILES command is used MS-DOS assumes a default value of 8. Any value higher than 20 serves no function.

3.4.8 LASTDRIVE command

The LASTDRIVE command is used to set the maximum drive letter which MS-DOS will accept.

The syntax of the LASTDRIVE command is:

LASTDRIVE=*d*

Where *d* is any letter from A to Z (and is case insensitive). When the drive letter is lower than the actual physical drives then MS-DOS ignores the LASTDRIVE specification and uses the default value which is the letter 'E'.

3.4.9 SHELL command

The SHELL command is used to specify an alternate top-level command processor in place of the standard COMMAND.COM file.

The syntax of the SHELL command is:

SHELL=*[<drive:>]<pathname> [param1 [param2... [param n]]]*

This command is used in conjunction with major software packages which furnish their own command processors. The MS-DOS technical manual contains information on developing command processors.

3.4.10 STACKS command

The STACKS command allows you to override the default MS-DOS stack resource parameters. For each hardware interrupt which occurs, MS-DOS allocates a stack to it, from the pool of available stacks. When the interrupt process is completed, MS-DOS returns the stack to the available stack pool.

The syntax of the STACKS command is:

STACKS=*<number of stacks> , <stack size in bytes>*

If there is no STACKS command in your CONFIG.SYS file then MS-DOS allocates default stack resources equivalent to the command STACKS=9,128. This however may not be sufficient if you are using multiple interrupting devices (such as LANs, 8087 NDPs or Hard Disks) and under these circumstances you may experience a number of stack related messages such as "Internal Stack Failure" (most predominantly) or even "Divide Overflow". When a message such as this occurs, it is advisable to try increasing the stacks, bearing in mind that stacks do use up available system memory in the same way that buffers and FCBs do. The number of stacks allowable is from 8 to 64 and the stack size parameter may vary from 32 to 512 bytes.

3.4.11 The supplied CONFIG.SYS file

The CONFIG.SYS file on the supplied MS-DOS floppy diskette contains the following:

```
COUNTRY=044,437
FILES=20
BUFFERS=20
```

If you use SELECT A: C: 044 UK to format a hard disk and install MS-DOS onto a hard disk then it will create a CONFIG.SYS file that just contains:

```
COUNTRY=044,437
```

It is a good idea to edit this file (using RPED for example) to add the FILES and BUFFERS commands because, as explained above, these will make the operation of MS-DOS more efficient.

3.5 AUTOEXEC.BAT File

When MS-DOS is first loaded into your PC, it looks for a text file called AUTOEXEC.BAT. MS-DOS expects this file to contain a sequence of commands which it should read as its first operational instructions.

Any of the MS-DOS commands described in Part 2 may be included in this file. Also, you could include the necessary commands to run your favourite application program or perhaps some combination of the two, so that they are automatically loaded every time MS-DOS is loaded into your PC.

Normally AUTOEXEC.BAT would at least contain the necessary KEYB command To instruct MS-DOS as to which type of keyboard is being used. It will, therefore, know which characters are represented by which keys. In the United Kingdom this command would be

KEYB UK 437

The supplied AUTOEXEC.BAT file contains the following commands:

```
PATH \:
KEYB UK 437
ECHO OFF
DATETEST
IF ERRORLEVEL 1 GOTO SIGNON
DATE
TIME
:SIGNON
CLS
VER
```

As described in Part 2, the PATH command instructs MS-DOS where to search for programs, the ECHO OFF command prevents the following commands from being shown on the screen as each is executed, the DATE and TIME commands are used to set the date and time, the CLS command clears the screen and the VER command prints the MS-DOS version number on the screen.

If you use SELECT A: C: 044 UK to format a hard disk and install MS-DOS onto a hard disk then it will create an AUTOEXEC.BAT file that just contains:

```
PATH C:\;
KEYB UK 437
ECHO OFF
CLS
DATE
TIME
VER
```

The extra lines in the supplied AUTOEXEC.BAT ensure that you are only asked to set the date and time when the stored date and time is not valid. Because your PC has a battery backed clock the date and time will normally be valid and these commands will not be executed.

After using SELECT, if desired, the extra lines could be added to AUTOEXEC.BAT using an editor such as RPED.

4. THE DISPLAY

Introduction

A video screen allows you to see the data you are typing in, the changes you are making to the data, and the end results of the data manipulation.

Software controls the layout of the data on the screen. There are two types of software:

- Software which gives an overall display of text and/or VGA, for example Windows

- Software which makes the screen work in electronic analogue mode: text is displayed from left to right and from top to bottom, in the same way that text is produced on an electronic typewriter. For example, MS-DOS allows you to type a command, such as DIR, to list the contents of a directory on the screen. The list is displayed with the first item written across the top line of the screen, from left to right, and the last item at the bottom of the screen.

4.1 The cursor

Both text-based and graphics-based software often use a cursor. It may be in the form of a flashing horizontal or vertical line, to attract your attention. A cursor provides a distinctive mark on the screen which indicates where the next item you type, will appear on the screen.

The cursor's movements are controlled so that the cursor is always in a sensible place: for example, if the cursor is at the top of the screen, and there are three paragraphs of text displayed, pressing the [↓] a number of times will probably make the cursor move from the beginning of one paragraph to the end of the paragraph; repeating the pattern of movements for each subsequent paragaph.

Using the keyboard and the mouse to control the cursor

There is usually a mechanism for moving the cursor around the screen: in most applications programs, you can use both the cursor keys [↑] [↓] [←] [→] and the mouse to move the cursor up and down , to the left and to the right, on the screen. The

mouse software and the mouse driver relate the movements of the mouse on the table to the movements of the cursor on the screen.

In text-based programs, you will probably find it easier to control the cursor using the cursor keys, whereas in graphics-based programs, the mouse is usually most useful.

Using just the mouse, or the [Enter], [Esc] and [F1] keys, will allow you to do a considerable amount of work: text-based software is often designed so that the [Esc], [Enter] and [F1] keys are very powerful keys. The [Enter] key usually takes you to a deeper level of the program, while [Esc] takes you back to the previous level of the program. Similarly, graphics-based software will often use the left mouse button to take you to a deeper level of the program.

Using the cursor and the pointer in graphics based applications

In graphics based applications, like Digital Research Gem, for example, there is a pointer as well as a cursor. In the same way that the mouse moves the rectangular cursor on a text screen, it moves the pointer on a graphics display. Pressing the mouse buttons issues instructions to act on the information on that part of the screen. In practice, this function is normally obvious. It is also possible to move the pointer using the cursor keys, though this is much slower than using the mouse.

As well as the mouse, the tab key [→|] can be used to move the cursor from one field to another.

In addition, a graphics program such as Windows will allow the tab key and the mouse to move the cursor while you are typing in text. For example, the Windows program uses "dialog" boxes to display a menu of options, leaving spaces where you are required to type in information, here you can use the mouse to move the pointer to select the option and then the tab key to move the cursor to the fields within the box.

4.2 Features of the mouse software

The MOUSE command supplied with your PC has various options which allow you to control the mouse's movements. You can use these options to:

- Make the mouse operate more smoothly by changing the clock speed.

- Assign key tokens to the mouse buttons. This means you can assign a character or function of your choice to a mouse button, rather than press the relative key on the keyboard. For example, the default settings are the [↵] key and the [Esc] key, so when

you press the left mouse button, the effect on the screen will be the same as if the [↵] key had been pressed and when you press the right mouse button, you will get the same effect as if you had pressed the [Esc] key.

- Set the scaling factor of the mouse. The scaling factor of the mouse is the distance the cursor moves across (or up and down) the screen, in relation to the distance the mouse is moved across (or up and down) the table. Adjusting the scaling factor of the mouse, allows you to determine the fierceness of the mouse movements. If you increase the scaling values, you can make the cursor go further on the screen with less movement of the mouse on the table.

See the MOUSE command in Chapter 6. for details of the options and how to use them.

NOTE Do not use the MOUSE command if you will be operating Windows, as it has its own built-in mouse software.

4.3 The Display Adaptor

Your PC contains a sophisticated multi-mode graphics display adaptor, called VGA (Video Graphics Array) as a built-in feature.

Any VGA analogue display may be used with your PC. These displays are available in monochrome, or color with a choice of screen sizes and resolution.

The resolution of color displays is expressed in fractions of a millimetre (e.g. 0.32mm). This measures the distance between the dots of the screen.

Your PC can produce **extended - VGA** which effects a higher resolution mode. To view this mode, you would need to use a high specification "multi-frequency" display.

4.4 Display specification and software compatibility

Use of color or monochrome software with VGA

The design of the latest graphics adaptor, VGA, allows software that was written for a monochrome display to be used with a color display; and software written for a color display with a monochrome display.

Earlier PCs also had a choice of monochrome or color displays, but software usually had to be chosen according to the type of display used: software written for a color display could not be used with a monochrome display and software written for monchrome would not work with color. If the wrong version of software were used, nothing would appear on the screen.

Therefore, most software was written to run on either a color or a monochrome display. The differences in design were not significant to text based programs but did affect graphics software.

However, VGA displays will work with all software, regardless of whether it was written for a color display or a monochrome display. Color software will appear as shades of grey on a monochrome display and monochrome software will give a black and white picture on a color display.

Two problems which might occur are:

- if you have color display and attempt to run monochrome software which has not been designed for use with a VGA adapter, you may find it necessary to issue the command MODE MONO.

- if you have mono display and attempt to run color software which has not been designed for use with a VGA adapter, you may find it necessary to issue the command MODE CO80.

See the technical information at the end of this chapter for more information.

Use of CGA, EGA, MDA or Hercules software with VGA.

The PC will also be able to run software written for the older EGA, CGA, MDA and Hercules graphics adaptors, irrespective of which VGA display is used. This is because the display adaptor has the ability to adapt to the mode of the software being used.

You will need to know what graphics adaptors your software can work with and then how to get your PC to work in that mode. Use the DEVICE command in Chapter 3.

You should not need to adjust any DIP switches or use any commands for this facility to work, but a special command is provided which you may have to use.

If you find that your software does not work, you may need to adjust DIP switch 6 from the default ON setting, to the OFF position, or use the MODE MONO command. See section 4.6 for technical information, or consult your dealer if you are in difficulty.

While the PC is in the default setting (DIP switch 5 in the OFF position and DIP switch 6 in the ON position) you should be able to use virtually any software on your PC that was written for a VGA display.

Note that a lot of software is installable, that is, although all software will work on your PC some will ask you to specify the type of display that you have. In which case, there will be a number of options. If you are given the choice of specifying a VGA display, then select that option, but if not the order of preference is: EGA, then Hercules, and finally, if you have none of the other alternatives, you will have to specify CGA.

4.5 Changing the graphics adaptor modes

The operation of the graphics adaptor is controlled by a combination of DIP switches and software commands. The default setting for the PC is with all the DIP switches in the OFF position, that is, switched up except switch 6 which will be in the ON position. The DIP switches are on the back panel of the system unit.

The software commands may be incorporated within your applications program and controlled automatically, or you may have to use the DISPLAY options of the DEVICE utility program, explained in Section 3.3.

4.6 Technical information - Video Display

Your PC contains a sophisticated multi-mode graphics adapter. Its operation is controlled by a combination of switches and software commands. Those software commands may be incorporated within your applications program, or they may be issued by the "DISPLAY" options of the DEVICE utility program.

Your PC will also cope with software written for the older EGA, CGA, MDA and Hercules graphics adapters; again, irrespective of the VGA monitor used. There are a few precautions, however, detailed below, to take in order to ensure complete adherence of all software to this scheme.

PS/2 or AT mode.

VGA is a graphics standard introduced as part of the architecture of the IBM PS/2 range of computers. These computers also introduced the MCA (Micro-Channel Architecture) expansion bus, thus precluding the possibility of having an additional CGA or MDA graphics adapter fitted. There is, however, a VGA adapter card available for the XT or AT type of computer. This has to allow for the possibility of a second non-VGA graphics adapter being fitted. The selection between PS/2 and AT mode is made by Display Selector switch 6. If in doubt PS/2 mode should be selected.

Switch 6: Normally ON to indicate PS/2 mode.

Monitor	Software	Action	Notes
Color	Color	Normal expected picture	
Color	Mono	Black and white picture	Some non-VGA s/w also requires MODE MONO command
Mono	Mono	Normal expected picture	
Mono	Color	Grey-scale picture	Some non-VGA s/w also requires MODE CO80 command

Switch 6: Optionally OFF to indicate AT mode.

Monitor	Software	Action	To operate..	Notes
Color	Color	Normal expected picture		
Color	Mono	Black and white picture	DEVICE DISPLAY MDA	then optionally
		either...	DEVICE DISPLAY VGA	VGA compatible monochrome s/w only
		..or	DEVICE DISPLAY EGA	EGA compatible monochrome s/w only
Mono	Mono	Normal expected picture		
Mono	Color	Grey-scale picture	DEVICE DISPLAY CGA	then optionally
		either...	DEVICE DISPLAY VGA	VGA compatible color s/w only
		..or	DEVICE DISPLAY EGA	EGA compatible color s/w only

This AT mode would normally be selected when either a secondary graphics adapter is to be used or software is exhibiting problems because it is making assumptions about the system that are not valid in PS/2 mode.

Certain additional graphics adapters are allowed in AT mode. An additional CGA, EGA or VGA is not permitted. The built-in VGA is always the Primary adaptor and the additional card the secondary. The following combination is the only one allowed:

Extra card+monitor	VGA monitor	S/W on VGA	Output to VGA	Output to Secondary Adapter.
Monochrome MDA or Hercules	Color	Color	MODE CO80	MODE MONO

NB. Hercules FULL mode is not permitted.

There are no switches to set. The configuration is automatic.

Disabling the built-in VGA.

It is possible to disable the built-in VGA and use one or more alternative graphics adapter(s) plugged into an expansion slot. The switch settings required on the graphics adapter (particularly if it is an EGA type) will be described in the user instructions for that card. If in doubt consult the dealer from whom you purchased it.

The switch settings required, if the Built-in VGA is disabled, are as follows:

Sw1	Sw2	Sw4	
OFF	OFF	ON	EGA or VGA
OFF	ON	ON	CGA 40 column
ON	OFF	ON	CGA 80 column
ON	ON	ON	MDA or Hercules
(x	x	OFF	Built-in VGA enabled)

Using "Hardware Compatible" modes : the DISPLAY command

Your PC is capable of switching to Hardware (register) compatibility modes for operating with software written for the older CGA, MDA and Hercules graphics adapters. Some of that software (particularly if it only displays text) will not require hardware compatibility. Other software will, and you have to change mode. This can be done by the user with the DEVICE DISPLAY <mode> command or by software (BIOS calls AH=00h, Al=xxh, INT 10h; AH=00h, AL=7Fh, BH=01h, INT 10h)

The valid modes to use with the DEVICE DISPLAY utility are CGA, MDA, HERC0 and HERC1. The BIOS modes with hardware compatibility available are 0 - 7.

EGA and MCGA BIOS-level compatibility are always provided.

Use with multi-frequency monitors.

Switch 5 may be used to tell your PC to use special timing for Multi-frequency monitors. Otherwise it will use standard VGA timing suitable for fixed-frequency analog monitors (and some newer multi-frequency monitors). This setting will allow most standard video modes to be displayed using a larger screen area on many brands of multi-frequency monitor. It is possible that you may need to experiment with this setting to determine the best results if you are using a multi-frequency monitor.

It is also possible to read the setting of this switch in order to determine if it is wise to change into a mode only supported on a multi-frequency monitor. The special 800*600 graphics mode is one such mode. The VGATEST utility program supplied with your PC will only offer a test of 800*600 graphics if you have indicated that a multi-frequency monitor is attached. There are also slight differences in the way in which 132 column text modes are handled.

To read the setting of the multi-frequency switch, a special BIOS sub-function is provided (BIOS call AH=00h, AL=7Fh, BH=0Fh, INT 10h)

Vertical resolution (lines).	Horizontal Frequency	Vertical Frequency	Modes
ff 350	+31.47 KHz	-70.08 Hz	0*,1*,2*,3*,7,F,10h
ff 400	-31.47 KHz	+70.08 Hz	0+,1+,2+,3+,7+
ff 200ds	-31.47 KHz	+70.08 Hz	0,1,2,3,4,5,6,D,E,13h
ff 480	-31.47 KHz	-59.94 Hz	11h,12h
ff 387/400 (132 col)	-31.5 KHz	+70.08 Hz	54h,55h,56h,57h (7 dot font)
ff 600	<---- not allowed on ff (fixed frequency) monitor		--->
mf 350	+31.5 KHz	-70.1 Hz	0*,1*,2*,3*,7,F,10h
mf 400	-27.97 KHz	+62.3 Hz	0+,1+,2+,3+,7+
mf 200ds	-31.47 KHz	+62.3 Hz	0,1,2,3,4,5,6,D,E,13h
mf 480	-31.47 KHz	-59.94 Hz	11h,12h
mf 387/400 (132 col)	-27.6 KHz	+61.5 Hz	54h,55h,56h,57h (8 dot font)
mf 600	-35.16 KHz	-56.16 Hz	58h,59h
mf 400 (extended)	-31.5 KHz	+62.3 Hz	5Eh

where:

*	indicates a EGA text mode with 8*14 or 9*14 character sizes
+	indicates a VGA text mode with 9*16 character sizes
ff	= Fixed frequency
mf	= multifrequency
ds	= double scanned (ie 200 line modes displayed double scanned)
"+"	indicates positive sync
"-"	negative sync

VGA modes (with EGA & MCGA BIOS compatibility)

Mode	Type	Colors	Columns	Rows	Buffer@	Characters	Resolution
0+	text/c	16/256k	40	25	B8000h	9*16	360*400
1+	text/c	16/256k	40	25	B8000h	9*16	360*400
2+	text/c	16/256k	80	25	B8000h	9*16	720*400
3+	text/c	16/256k	80	25	B8000h	9*16	720*400
4	graph/c	4	40	25	B8000h	8*8	320*200ds
5	graph/c	4	40	25	B8000h	8*8	320*200ds
6	graph/c	2/256k	80	25	B8000h	8*8	640*200ds
7+	text/m	4	80	25	B0000h	9*16	720*400
D	graph/c	16/256k	40	25	A0000h	8*8	320*200ds
E	graph/c	16/256k	80	25	A0000h	8*8	640*200ds
F	graph/m	4	80	25	A0000h	8*14	640*350
10	graph/c	16/256k	80	25	A0000h	8*14	640*350
11	graph/c	2/256k	80	30	A0000h	8*16	640*480
12	graph/c	16/256k	80	30	A0000h	8*16	640*480
13	graph/c	256/256k	40	25	A0000h	8*8	320*200ds

Extended VGA modes (Multi-frequency monitor)

Mode	Type	Colors	Columns	Rows	Buffer@	Characters	Resolution
54h	text/c	16/256	132	43	B8000h	8*9	1056*387
55h	text/c	16/256	132	25	B8000h	8*16	1056*400
56h	text/m	4	132	43	B0000h	8*9	1056*387
57h	text/m	4	132	25	B0000h	8*16	1056*400
58h	graph/c	16/256k	100	75	A0000h	8*8	800*600
59h	graph/m	2	100	75	A0000h	8*8	800*600
5Eh	graph/c	256/256k	80	25	A0000h	8*16	640*400

Extended VGA modes (Fixed-frequency monitor)

Mode	Type	Colors	Columns	Rows	Buffer@	Characters	Resolution
54h	text/c	16/256	132	43	B8000h	7*9	924*387
55h	text/c	16/256	132	25	B8000h	7*16	924*400
56h	text/m	4	132	43	B0000h	7*9	924*387
57h	text/m	4	132	25	B0000h	7*16	924*400
5Eh	graph/c	256/256k	80	25	A0000h	8*16	640*400

NB. Modes 58h and 59h can be selected, but will probably not be displayed correctly due to inherent limitations of the fixed-frequency monitor.

CGA Hardware compatible modes

Mode	Type	Colors	Columns	Rows	Buffer@	Characters	Resolution
0	text/c	16/256K	40	25	B8000h	8*8	320*200ds
1	text/c	16/256K	40	25	B8000h	8*8	320*200ds
2	text/c	16/256K	80	25	B8000h	8*8	640*200ds
3	text/c	16/256K	80	25	B8000h	8*8	640*200ds
4	graph/c	4	40	25	B8000h	8*8	320*200ds
5	graph/c	4	40	25	B8000h	8*8	320*200ds
6	graph/c	2/256	80	25	B8000h	8*8	640*200ds

MDA/Hercules Hardware compatible modes

Mode	Type	Colors	Columns	Rows	Buffer@	Characters	Resolution
7	text/m	4	80	25	B0000h	9*14	720*350
Herc0	graph/m	4	80	25	B0000h	9*14	720*348
Herc1	graph/m	4	80	25	B0000h	9*14	720*348
					..plus B8000h		

Special considerations in 132 column text modes.

Your PC has an 8 pixel-wide font supplied in ROM for the 132 column modes. However, if used with a fixed-frequency monitor (determined by the setting of Switch 5), only the first 7 pixels are displayed. If using the 132 column modes with a fixed-frequency monitor it may be desirable to down-load a more suitable 7 pixel-wide font and select the alternate print screen routine. Suitable fonts will have to be supplied as binary files - 7*9 for 43 row, 7*16 for 25 row.

Summary of Video switch settings

Sw1	Default Mode select when Built-in VGA disabled
Sw2	Default Mode select when Built-in VGA disabled
(Sw3	disable Serial port COM1)
Sw4	Disable built-in VGA
Sw5	Multi-frequency monitor attached
Sw6	ON = PS/2 mode selected; OFF = AT mode selected
Sw7	Reserved
Sw8	Reserved

The standard screen handling provided by MS-DOS is just a very simple teletype mode.

Characters with internal codes in the range 32..255 (020h..0FFh) are displayed on the screen at the current cursor position. Generally the cursor is then moved right by one column. If, however, the cursor is at the column furthest to the right and wrapping is enabled, it will move to the column furthest to the left on the next line, scrolling the screen up if necessary. If wrapping is not enabled, the final character on the line will be overwritten.

Characters with internal codes in the range 0..31 (00h..01Fh) are treated as control codes as follows:

7 (07h) **BEL** (Bell): Sounds a bleep

8 (08h) **BS** (Backspace): Moves the cursor one column to the left. If the cursor is at the column furthest to the left and wrapping is enabled, it is moved to the column furthest to the right on the row above - unless it is already on the top row.

10 (0Ah) **LF** (Line Feed): Moves the cursor down one line; scrolling the screen up if necessary.

13 (0Dh) **CR** (Carriage Return): Moves the cursor to the column furthest to the left on the present row.

27 (1Bh) **ESC** (Escape): Introduces an escape sequence.

All other control codes are ignored.

If the CONFIG.SYS file (see Chapter 3.4) is changed to include the line:

DEVICE=ANSI.SYS

then extended screen handling facilities become available. These use standard ANSI screen control codes. These codes are as follows:

ESC [*n***A** Moves the cursor up *n* rows, unless it is already at the top of the screen. If *n* is omitted, the value 1 is assumed.

ESC [*n***B** Moves the cursor down *n* rows, unless it is already at the bottom of the screen. If *n* is omitted, the value 1 is assumed.

ESC [*n***C** Moves the cursor right *n* columns, unless it is already in the column furthest to the right. If *n* is omitted, the value 1 is assumed.

ESC [nD Moves the cursor left *n* columns, unless it is already in the column furthest to the left. If *n* is omitted, the value 1 is assumed.

ESC [*n*;*m*f	Move the cursor to row *n*, column *m*. If *n* or *m* is omitted the value 1 is assumed.
ESC [*n*;*m*H	Move the cursor to row *n*, column *m*. If *n* or *m* is omitted the value 1 is assumed.
ESC [=*n*h	Set screen width and type (see Table 1, below)
ESC [2J	Clear the screen and return the cursor to its home position.
ESC [K	Erase to the end of the line.
ESC [=*n*l	Reset screen width and type (see Table 1, below).
ESC [*n*;...;*k*m	Set graphics parameters (see Table 2, below).
ESC [6n	Report cursor position to the system.
ESC [*sic*;*parm*[;*parm*...]p	Map standard internal code associated with a key, *sic*, onto another code or sequence of codes, thereby redefining the keyboard. *parm* is a decimal number or string.
ESC [*n*;*m*R	Specify current cursor position as row *n*, column *m* and report this position to the system.
ESC [s	Save the cursor position.
ESC [u	Restore the cursor position.

Table 1: Screen width and type ESC [=*n*h and ESC [=*n*l

n=0	40x25 black and white
n=1	40x25 color
n=2	80x25 black and white
n=3	80x25 color
n=4	320x200 color
n=5	320x200 black and white
n=6	640x200 black and white
n=7	wrap at end of line

Table 2: Graphics parameters ESC [*n*;...;*k*m

n..k take values as follows:

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
48	Subscript	49	Superscript

30	Black foreground	40	Black background
31	Red foreground	41	Red background
32	Green foreground	42	Green background
33	Yellow foreground	43	Yellow background
34	Blue foreground	44	Blue background
35	Magenta foreground	45	Magenta background
36	Cyan foreground	46	Cyan background
37	White foreground	47	White background

NOTE: The functionality of ANSI.SYS will only be supported in true VGA modes.

Programming the built-in VGA

Your Disk Operating System (and if used, Graphics Environment or Presentation Manager) provides a standard interface to the screen. Consult the reference publications for that software. Many programmers, however, prefer more direct access to the graphics adapter; or require facilities not normally otherwise provided.

The ROM BIOS includes routines for reading and writing the screen. Programmers should consult the relevant Industry Standard publications for details of the video hardware (including palettes and bitwise memory mapping) in VGA, EGA, CGA and MDA modes.

Light-pen functionality is only available if a "real" CGA card is installed.

Interrupt 16 (int 10h): 'VGA Compatible' Video I/O.

The BIOS interrupt 16 service routine provides an extended set of service routines similar in many respects to those of a true 'XT' compatible BIOS. The extensions support enhancements in the graphics hardware environment and are not available (the ROM is literally switched out) if the built-in VGA is disabled.

Video Int 16 Function 0: Set Video Mode.

Entry: AH = 0

AL = Mode required.

00h-07h & 0Dh-13h	VGA compatible
54-5Fh	Extended VGA
7Eh	Parametric mode set
7Fh	Switch VGA/Hardware compatible modes

For mode 7Eh:

BX = Horiz res. pixels or columns
CX = Vert res. pixels or columns
DX = No. colors

For mode 7Fh:

BH = 00h Set VGA compatible mode (note 1)
BH = 01h Set H/W compatible mode (note 1)
BH = 02h Return extended mode status
BH = 03h Lock mode for reset (note 2)
BH = 0xh Read Extended VGA register 'x' (09h-0Fh)
BH = 1xh Write Extended VGA register 'x' (09h-0Fh)

Exit:

For mode 7Fh:	BH = 00h - 03h:	BH = 7Fh if call was valid
	BH = 02h	BL = as follows: 00h if VGA compatible mode 01h if H/W compatible mode CH = memory fitted on adapter (note 3) CL = memory required for mode (note 3)
	BH = 09h - 0Fh:	BH = 7Fh if call was valid BL = value read (note 4)
	BH = 19h - 1Fh:	BH = 7Fh if call was valid

For all other modes: All flags and registers preserved.

Sub-function 7Eh derives the mode required by 'best fit' of the supplied parameters.

1 The graphics adapter retains the previously set mode; this implies that when switching to hardware compatiblity, a mode in the range 00h-07h has been previously selected.
2 Unless this call is made, a reset will switch the graphics adapter to a VGA compatible mode - 07h for monochrome monitors, 03h color monitors.

3 Memory in 64k units.

4 Register 0Fh contains a copy of four of the Display Selector switches. Bit 7 = Switch 5; Bit 6 = Switch 6; Bit 5 = Switch 7; Bit 4 = Switch 8.

Video Int 16 Function 1: Set Cursor Size.

Entry: AH = 01h
CH = Starting scan of cursor
CL = Ending scan of cursor

Exit: All flags and registers preserved.

Values of CH from 20h to 3Fh turn the cursor off. Setting cursor start greater than cursor end usually turns the cursor off and other combinations give strange cursors. This function is only relevant in alpha modes as the cursor is not supported in graphics modes. It sets the CRTC cursor start and cursor end values.

Video Int 16 Function 2: Set Cursor Address.

Entry: AH = 02h
BH = Page number
DH = Cursor Row Address
DL = Cursor Column Address

Exit: All flags and registers preserved.

Video Int 16 Function 3: Get Cursor Address.

Entry: AH = 03h
BH = Page Number

Exit: DH = Cursor Row Address
DL = Cursor Column Address
CH = Starting scan of cursor
CL = Ending scan of cursor

All flags and other registers preserved.

Video Int 16 Function 4: Get Light Pen Address.

Entry: AH = 04h

Exit: If Light Pen switch set then
AH = 01h
DH = Character Row address (0 to 24)
DL = Character Column address (0 to 79)
CH = Pixel Row address (0 to 199) for modes 0 - 07h
CL corrupt if modes 0 - 07h
CX = Pixel Row address (0 to 349) for modes 0Dh - 10h

BX = Pixel Column address (0 to 639)
If Light Pen switch clear then
AH = 0
BX, CH & DX preserved.
Always
All flags and other registers preserved.

Video Int 16 Function 5: Set Display Page.

Entry: AH = 05h
AL = Page Number to be selected

Exit: All flags and registers preserved.

Video Int 16 Function 6: Scroll Screen Up.

Entry: AH = 06h
DH = Bottom Row of area to scroll
DL = Right most Column of area to scroll
CH = Top Row of area to scroll
CL = Left most Column of area to scroll
BH = Attributes for blank lines scrolled onto the bottom.of the scroll area
AL = Number of lines to roll up
If AL = 0 then blank the specified area
If AL not zero then roll specified area up by the number of lines in AL
If DH = CH then AL must be zero

Exit: All flags and registers preserved.

The attribute byte is defined as:

Bit	Definition
7	Intensity or Enable Blink (Background)
6	Red (Background)
5	Green (Background)
4	Blue (Background)
3	Intensity (Foreground)
2	Red (Foreground)
1	Green (Foreground)
0	Blue (Foreground)

Video Int 16 Function 7: Scroll Screen Down.

Entry: AH = 07h
DH = Bottom Row of area to scroll
DL = Right most Column of area to scroll
CH = Top Row of area to scroll
CL = Left most Column of area to scroll
BH = Attributes for blank lines scrolled onto the top of the scroll area
AL = Number of lines to roll down
If AL = 0 then blank the specified area
If AL not zero then roll specified area down by the number of lines in AL
If DH = CH then AL must be zero

Exit: All flags and registers preserved.

Video Int 16 Function 8: Read Character and Attributes (at cursor position).

Entry: AH = 08h
BH = Page to read

Exit: AL = Character
AH = Attributes byte. (Unchanged in graphics modes)

All flags and registers preserved.

Video Int 16 Function 9: Write Character and Attributes (at cursor position).

Entry: AH = 09h
AL = Character to write
CX = Repeat Count
BH = Page to write
BL = In alpha modes
Attributes of character
In graphic modes
Write mode and character color:
Bit 7 = 0 for character overwrite mode
Bit 7 = 1 for character XOR mode
Bits 0 to 3 = character color

Exit: All flags and registers preserved.

Video Int 16 Function 10: Write Character Only (at cursor position).

Entry: AH = 0Ah
AL = Character to write
CX = Repeat Count
BH = Page to write
BL = In alpha modes
BL is not used
In graphic modes
Write mode and character color:
Bit 7 = 0 for character overwrite mode
Bit 7 = 1 for character XOR mode
Bits 0 to 3 = character color data

Exit: All flags and registers preserved.

Video Int 16 Function 11: Write Color Palette.

Entry: AH = 0Bh
BH = Function select:

Zero	Set background color as specified by BL.
Non Zero	Set the Palette (0 or 1) as specified by BL.

Exit: All flags and registers preserved.

The palette number setting (BH non-zero) only applies to 320 x 200 graphics mode.

Video Int 16 Function 12: Write a Pixel.

Entry: AH = 0Ch
DX = Pixel Row
CX = Pixel Column
AL = Write Mode & Pixel:
Bit 7 = 0 for character overwrite mode
Bit 7 = 1 for character XOR mode
Bits 0 to 3 = Pixel Data

Exit: All flags and registers preserved.

The pixel data specified in AL should be in the range 0 to 3 in 320 x 200 graphics modes, in the range 0 to 1 for mode 640 wide B/W graphics modes,and the range 0 to 0Fh for 640 wide 16 color graphics modes. This function is not supported for Extended VGA modes.

Video Int 16 Function 13: Read a Pixel.

Entry: AH = 0Dh
DX = Pixel Row
CX = Pixel Column

Exit: AL = Specified pixel's color
All other registers preserved.
Flags preserved.

This function is not supported for Extended VGA modes.

Video Int 16 Function 14: Write in TTY Emulation Mode (at cursor position).

Entry: AH = 0Eh
AL = Character to Write
BL = In alpha modes
BL is not used
In graphics modes
Write mode & Pixel Data:
Bit 7 = 0 for character overwrite mode
Bit 7 = 1 for character XOR mode
Bits 0 to 3 = character color

Exit: All flags and registers preserved.

The following characters are executed rather than displayed symbolically:

BEL (07h)	Sounds a short (bleep) note on the speaker.
BS (08h)	Decrements the cursor column one character position unless the column is already zero in which case it is ignored.
CR (0Dh)	Sets the cursor column address to zero.
LF (0Ah)	Increments the cursor row address by one and if then the row is greater than the screen height the screen is scrolled up a line and the row is decremented.

All other characters are displayed and the cursor position is incremented by one.

Video Int 16 Function 15: Get Current Video Parameters.

Entry: AH = 0Fh

Exit: AH = Number of visible columns (40 or 80).
AL = Current Video Mode
BH = Active Display Page Number

All other registers preserved.
Flags preserved.

Video Int 16 Function 16: Set Palette Register(s).

Entry: AH = 10h
AL = Sub-Function Selector as follows:
0 - Set Individual Palette Register
1 - Set Hi-Res Border Register
2 - Set All Palette Registers and Hi-Res Border Register
3 - Set Blink/Intensity Mode Bit
10h - Set Individual Color Register
12h - Set Block of Color Registers
15h - Read Individual Color Register
17h - Read Block of Color Registers
1Bh - Sum Colors to Grey Scale

Sub-function 0: BL = Palette register number (0-0Fh).
BH = Palette Value (0-3Fh).

Sub-function 1: BH = Hi-Res Border Value (0-3Fh).

Sub-function 2: ES:DX = Pointer to a 17 byte table as follows:
0 to 15 = Palette Registers 0-0Fh Value.
16 = Hi-Res Border Value.

Sub-function 3: BL = 1 - Enable Blink (Disable Intensified Foreground).
0 - Disable Blink (Enable Intensified Foreground).

Sub-function 10h: BX = Color Register to Set
DH = Red value (0-3Fh)
CH = Green value (0-3Fh)
CH = Blue value (0-3Fh)

Sub-function 12h: ES:DX = Pointer to table of Red/Green/Blue values
BX = First color register to set (0-FFh)
CX = Number of color registers to set

Sub-function 15h: BX = Color Register to read

Sub-function 17h: ES:DX = Pointer to space for table
BX = First register to read (0-FFh)
CX = Number of registers to read

Sub-function 1Bh: BX = First color register to sum
CX = Number of color registers to sum

This sub-function reads in color registers, sums them as 30% Red, 59% Green, 11% Blue and writes the result back to Red, Green and Blue registers. The original data is therefore destroyed.

Exit:

Sub-function 15h: DH = Red value
CH = Green Value
CL = Blue value

Other functions: All flags and registers preserved.

Sub-functions 0-2 are primarily intended for use with EGA compatible software. Sub-functions 10h-1Bh are intended for use with VGA and extended VGA modes only.

VGA Video Int 16 Function 17: Load Character Generator.

This function enables setting of extended character sets.

Entry: AH = 11h
AL = Sub-Function Selector as follows:
00h - Load User Table
01h - Load ROM 8x14 Font Block
02h - Load ROM 8x8 Font Block
03h - Set Character Set Select Register
10h - Load User Table
11h - Load ROM 8x14 Font Block
12h - Load ROM 8x8 Font Block
20h - Set Video 8x8 Matrix Table (Int Vect 31)
21h - Set User Extended Hi-Res Matrix Vector & Matrix Size
22h - Set ROM 8x8 Extended Hi-Res Matrix Vector & Size
23h - Set ROM 8x14 Extended Hi-Res Matrix Vector & Size
30h - Return Character Set Vector

Sub-function 0: BL = Font Block to load (0-3)
DX = Starting Offset into Font Block (0-255 characters)
BH = Points (Bytes per Character) (1-31)
CX = Number of Characters to load (1-256)
ES:BP = User Font Table Address Pointer

Sub-function 1: BL = Font Block to load (0-3)

Sub-function 2: BL = Font Block to load (0-3).

Sub-function 3: BL = 4-bit Character Sel A/B setting.

Sub-function 10h: (No additional parameters)

Sub-function 11h: BL = Font Block to load (0-3).

Sub-function 12h: BL = Font Block to load (0-3).

Sub-function 20h: ES:BP = User Graphics Font Address Pointer for characters 128 to 255.

Sub-function 21h: ES:BP = User Graphics Font Address Pointer for characters 0 to 255.
CX = Bytes per Character. (1-31)
BL = 0 - Rows is specified by DL.
1 - Rows = 14.
2 - Rows = 25.
3 - Rows = 43.

Sub-function 30h: BH = as follows:
0 - Return vector 31 (User 8x8 upper 128 vector)
1 - Return vector 67 (Extended Hi-Res 256 vector).
2 - Return 8x14 ROM font address.
3 - Return 8x8 ROM font address.
4 - Return upper 128 origin of 8x8 ROM font address.
5 - Return 9x14 ROM font address.

Exit:

Sub-function 30h:
CX = Bytes Per Character (Points from 0:485).
DL = ROWS - 1 (from 0:484).
ES:BP = Requested Vector.

Other functions:
All flags and other registers preserved.

Video Int 16 Function 18: Return EGC state/Set PrtSc Vector.

Entry: AH = (12h)
BL = 10h - Return EGC Status.
20h - Set PrtSc Vector to the VGA ROM BIOS print screen entry.

Exit: If BL= 10h then registers returned as follows:
BH = 0 - Color Mode Set.
1 - Monochrome Mode Set.
BL = Number of 64Kb Video Ram Blocks less one.
CL = Switch (1 to 4) states (inverted - from 0:488).
CH = Features (0Fh - Features connector normally installed).

All flags and other registers preserved.

If BL = 20h then all flags and registers are preserved.

VGA Video Int 16 Function 19: Write String.

Entry: AH = 13h
AL = String Function Selector as follows:

0 - String consists of characters only and attribute in BL.
The Cursor stays at the position prior to the string write.

1 - String consists of characters only and attribute in BL.
The Cursor is updated to be the character after the string.

2 - String consists of character and attribute pair(s).
The Cursor stays at the position prior to the string write.

3 - String consists of character and attribute pairs(s).
The Cursor is updated to be the character after the string.

BH = Page to Write.
DH = Cursor Row Address.
DL = Cursor Column Address.
CX = Character Count (not including attribute bytes if supplied with the string).
ES:BP = Address Pointer to the String.

Exit: All flags and registers preserved.

VGA Video Int 16 Function 26: Read/Write display combination

Entry: AH = 1Ah select Read/Write Display Combination function
AL = Sub-function selector as follows:
0 - Read Display Combination Code
1 - Write Display Combination Code

Sub-function 0: BL = Active Display Code
BH = Alternate Display Code

Sub-function 1: BL = Active Display Code
BH = Alternate Display Code

Display Codes used are: 0 = 'No display'; 1 = 'Monochrome'; 2 = 'Color'.

VGA Video Int 16 Function 27: Return function and Video State

Entry: AL = 1Bh
BX = 0
ES:DI = Returned data buffer

Exit: ES:DI = Requested data

The format of the information returned is as follows:

Offset	Length	Contents
00h	2	Offset to Static Functionality Information
02h	2	Segment to Static Functionality Information
04h	1	Video Mode
05h	2	Columns per screen
07h	2	Length of Video Buffer in bytes
09h	2	Starting address of Video Buffer
0Bh	10h	Cursor Position for 8 display pages (Row/Column)
1Bh	2	Cursor Mode Setting (Cursor Start/End Value)
1Dh	1	Active Display Page
1Eh	2	CRT Controller address (3B4h or 3D4h)
20h	1	Value of CRT_MODE_SET
21h	1	Value of CRT_PALETTE
22h	1	Number of rows on the screen
23h	2	Character Height - number of scan lines per char
25h	1	Display Combination Code for Active Display
26h	1	Display Combination Code for Alternate Display
27h	2	Number of color supported for current video mode
29h	1	Number of display pages supported for current mode
2Ah	1	Code for Number of scan lines in current mode 0 = 200 Scan lines 1 = 350 Scan lines 2 = 400 Scan lines 3 = 480 Scan lines
2Bh	1	Primary Active Character Block (0-0xFFh)
2Ch	1	Secondary Active Character Block (0-0xFFh)
2Dh	1	Miscellaneous State Information 01h = All modes on all monitors are active 02h = Summing to Grey Scale is enabled 04h = Monochrome Display is attached 08h = Disable Default Palette Loading 10h = Cursor Emulation is Active 20h = Background intensity(0)/Blinking(1)
2Eh	3	Reserved
31h	1	Code for Amount of Video Memory available 0 = 64K 1 = 128K

Offset	Length	Contents
		2 = 192K 3 = 256K
32h	1	Save Pointer State Information 01h = 512 Character Set active 02h = Dynamic Save Area Active 04h = Alpha Font Override Active 08h = Graphics Font Override Active 10h = Palette Override Active 20h = DCC Extension Active
33h	0Dh	Reserved

The Static Functionality Table has the following format:

Offset	Length	Contents
00h	1	Bit Map of Video modes supported 01h = Mode 00h 02h = Mode 01h 04h = Mode 02h 08h = Mode 03h 10h = Mode 04h 20h = Mode 05h 40h = Mode 06h 80h = Mode 07h
01h	1	Bit Map of Video modes supported (continued) 01h = Mode 08h 02h = Mode 09h 04h = Mode 0Ah 08h = Mode 0Bh 10h = Mode 0Ch 20h = Mode 0Dh 40h = Mode 0Eh 80h = Mode 0Fh
02h	1	Bit Map of Video modes supported (continued) 01h = Mode 10h 02h = Mode 11h 04h = Mode 12h 08h = Mode 13h F0h = Reserved
03h	4	Reserved
07h	1	Number of Scan lines available in alpha modes 01h = 200 Scan lines 02h = 350 Scan lines 04h = 400 Scan lines F8h = Reserved
08h	1	Number of Character Set Blocks available in alpha modes

Offset	Length	Contents
09h	1	Maximum number of active Character Set Blocks available
0Ah	1	Miscellaneous State Information 01h = All modes on all monitors are active 02h = Summing to Grey Scale is enabled 04h = Character Font Loading 08h = Disable Default Palette Loading 10h = Cursor Emulation is Active 20h = EGA Palette 40h = Color Registers 80h = Color Paging
0Bh	1	Miscellaneous State Information (continued) 01h = Light Pen 02h = Save/Restore 04h = Background Intensity/Blinking control 08h = DCC F0h = Reserved
0Ch	2	Reserved
0Eh	1	Save Pointer State Information 01h = 512 Character Set active 02h = Dynamic Save Area Active 04h = Alpha Font Override Active 08h = Graphics Font Override Active 10h = Palette Override Active 20h = DCC Extension Acitve C0h = Reserved
0Fh	1	Reserved

Programming Extended Modes (54h-5Eh)

Extended VGA text modes may be accessed via the ROM BIOS in the normal way. Extended VGA graphics modes 58h, 59h and 5Eh have no BIOS support; the hardware is described below.

WARNING: Use of the Extended modes with certain Applications Programs, Graphics Environments or Presentation Managers will require special device drivers. The information below is published to facilitate the implementation of graphics drivers, or the use of the Extended graphics modes in stand-alone applications. Extended text modes, particularly those with more than 25 lines of text, may require re-installation of applications software.

	58h,59h	**5Eh**
Bankswitching	Same as EGA/VGA mode 10h	Algorithm below
Video RAM	Same as VGA, extended size	Bank-switched 4*64k

Mode 5Eh memory organization:

The display buffer organized as a number of 4k screen banks. Any sixteen contiguous banks can be switched into the 64k address space A0000h - AFFFFh. The number of the starting bank is set into a special register. Each byte holds one pixel (a palette of 256 colors is available).

A convenient algorithm for addressing any pixel is: If the Effective Address (EA) of a pixel (x,y), where the origin is at the top left hand corner of the screen, is defined as (y*640) + x.

Then Bank = INT(EA/64k) * 16
Address = (EA mod 64k) + A0000h

To bring the required banks into the address space, special register pairs need to be addressed. This is done by the following instructions, which must be used exactly as shown:

```
OUT 3CEh,0Fh
OUT 3CFh,05
OUT 3CEh,09
OUT 3CFh,bank
OUT 3CEh,0Fh
OUT 3CFh,00
```

5. THE KEYBOARD

The keyboard is used to enter two types of information into your PC: data which is to be stored and manipulated, for example letters, reports and accounts; and instructions which tell the computer how to process the data, for example, MS-DOS command lines. Both types of information require the use of character keys and numeric keys.

To enable the keyboard

Unlock the security switch on the lefthand panel of the system unit.

Functions of the keys

Character keys	In the centre of the keyboard, the character keys are organised in the QWERTY layout, that is, the most usual typewriter layout. Press to enter lower case characters (**a - z**) and the character on the lower half of the key, where there are two. Use the shift key [⇑] to enter upper case characters (**A - Z**) or the character on the upper half of the key.
[Caps Lock]	A toggle-switch key. Press once to produce upper case letters with the character keys; press once again to return to producing lower case letters with the character keys.
[⇑]	Hold down whilst pressing character keys, A - Z, to produce upper case letters; hold down whilst pressing **[Alt]**, **[Ctrl]** and **[PrtSc]** to give special characters or computer codes.
[Ctrl] [Alt]	**Ctrl** stands for Control; **Alt** stands for Alternative. Hold down whilst pressing other keys to give special characters or special computer codes.
Space bar	Enters a space character.
[Num Lock]	A toggle-switch key. Press this key once to switch to and from using the **[Home]** **[End]** **[Pg Up]** **[Pg Dn]** and the cursor keys, to using the numbers on the numeric keypad.
[Home] [End] [Pg Up] [Pg Dn]	These keys are used by text editing programs to move the cursor to the beginning of a document, the end of a document, one page further up in the document, and one page further down.

[↑] [↓] [←] [→]
Cursor keys

Pressing these four keys moves the cursor on the screen in the direction shown by the arrow. These keys are disabled by pressing the **[Num Lock]** key.

Note that there are two sets of cursor keys on the keyboard: one set is on the numerical keypad and is disabled when **[Num Lock]** is pressed,the second set, to the left of the numerical keypad, can be used at all times.

[←Del]

This key deletes the character to the left of the cursor.

[Del→]

Deletes the character under/to the right of the cursor.

[Del]

Use like **[←Del]** to delete the character immediately before/under the cursor. Also used with **[Ctrl]** and **[Alt]** to reset the PC.

[Esc]

Esc stands for escape. Whilst using programs, you can press the **[Esc]** key to return to the previous level of the program, and eventually to exit the program.

[Ins]

Stands for insert. A toggle-switch key. Press once to switch to overwriting existing text. Press once to return to inserting text.

[Scroll lock]

This key is for use with word processors. It enables the cursor keys to move the text on the screen behind the cursor, rather than move the cursor over the text. With some programs it has no effect.

[Break]

Use with the **[Ctrl]** key to abandon the program that is currently being run. Note that **[Break]** will sometimes abort everything you are using, and with some programs, will not work at all.

[Pause]

Pressing **[Pause]** once will stop the program you are using. Pressing any other key will then continue.

[PrtSc]

Use this key to print out an image of the screen on your printer. It must not be pressed if there is not a printer connected to your PC.

[Sys Rq]

Use this key by holding down **[Alt]** and pressing **[Prt Sc]**. The key may be used by an operating system to switch back to command level.

Function keys
[F1] [F2] etc.

These keys are defined with specific actions for individual programs. That is, each operating system and each application program will assign specific functions to each key. It is unlikely that these functions will be the same from program to program.

[→|]	The Tab key moves the cursor to the next Tab stop, inserting spaces as necessary.
[↵]	Pressing the Return key has two effects. One places the cursor on a new line, like the carriage return key of a typewriter; the other enters information you have just typed, sending an instruction to your computer. The effect this key has, depends on the context in which it is used: if you press this key whilst typing a letter, using a word processor, the cursor will simply move to the beginning of the next line; if you type an MS-DOS command line for example, you must then press the [↵] key to make MS-DOS read the instruction you have typed.
[Enter]	The Enter key functions in the same ways as the [↵] key; when using the numeric keypad, it is easier to use than the [↵]. Note that some programs give this key and the [↵] key separate functions.
[-]	Minus key - used to produce a minus sign character. In some software it will subtract figures.
[+]	Plus key - used to produce a plus sign character. In some software it will add figures together.

Notes on using the keyboard

The number **0**, on the top row of the keyboard between **9** and **-**, is not interchangeable with the letter **O** between the characters **P** and **I**, even though these characters look the same.

Not all programs use the keys in the same way. For example, **[←Del]** sometimes just moves the cursor back one position; **[Del]** sometimes deletes the character under/to the right of the cursor. With some programs, the cursor keys and the **[Home]** **[End]** **[PgUp]** **[Pg Dn]** **[→|]** **[Ins]** and **[Break]** keys may have no effect.

The numeric keypad, which is enabled by pressing the **[Num Lock]** key, has a similar layout to an adding machine, with numeric keys, [+] and [-] keys and an **[Enter]** key. It is therefore useful for fast entry of figures.

The cursor keys, which are enabled by releasing the **[Num Lock]** key, were added to the keyboard when the need for them became apparent. Before then, an application program would control the cursor's movements by the use of, for example, the **[Ctrl]** key with the **[E]** **[S]** **[D]** **[X]** keys. The positions of these keys on the keyboard indicates the direction the cursor should move:

```
    E
S       D
    X
```

To move the cursor further in these directions the following keys were used:

```
    R
A       F
    C
```

Combinations of keys

Combinations of keys are used to provide many extra functions. The keys used are **[Ctrl]** and **[Alt]** pressed together with a character key or a function key. This means that although your keyboard has 102 keys, it is able to effect more than 102 functions through various combined keystrokes. Operating systems and application programs all assign different functions to these combined keystrokes: the meaning can change from program to progam.

However, some programs will use keystrokes that are used by other programs, so it is useful to be aware of patterns in key usage. You may find mnemonical uses of keys, for example, You may also find that some keys are relevant in some way to the functions, for example pressing the [b] key for a backwards movement.

Your keyboard can also produce more characters than those marked on the keys. Although the keyboard appears to offer 72 characters, as on a usual QWERTY keyboard, it can actually produce 256 characters, that is 184 more than are marked on the keyboard. The following table shows all the characters that can be produced, using the United States code page. (This is the default setting for your keyboard). For an explanation of code pages and how to use them, see section 5.1 'Code pages'.

Table of all characters

DEC		16	32	48	64	80	96	112	128	144	160	176	192	208	224	240
	Hex Digit	-0	-1	-2	-3	-4	-5	-6	-7	-8	-9	-A	-B	-C	-D	-E	-F
0	-0		►		0	@	P	`	p	Ç	É	á	▒	└	╨	α	≡
1	-1	☺	◄	!	1	A	Q	a	q	ü	æ	í	▓	┴	╤	ß	±
2	-2	☻	↕	"	2	B	R	b	r	é	Æ	ó	▒	┬	╥	Γ	≥
3	-3	♥	‼	#	3	C	S	c	s	â	ô	ú	│	├	╙	π	≤
4	-4	♦	¶	$	4	D	T	d	t	ä	ö	ñ	┤	─	╘	Σ	⌠
5	-5	♣	§	%	5	E	U	e	u	à	ò	Ñ	╡	┼	╒	σ	⌡
6	-6	♠	▬	&	6	F	V	f	v	å	û	ª	╢	╞	╓	µ	÷
7	-7	•	↨	'	7	G	W	g	w	ç	ù	º	╖	╟	╫	τ	≈
8	-8	◘	↑	(	8	H	X	h	x	ê	ÿ	¿	╕	╚	╪	Φ	°
9	-9	○	↓	)	9	I	Y	i	y	ë	Ö	⌐	╣	╔	┘	Θ	∙
10	-A	◙	→	*	:	J	Z	j	z	è	Ü	¬	║	╩	┌	Ω	·
11	-B	♂	←	+	;	K	[	k	{	ï	¢	½	╗	╦	█	δ	√
12	-C	♀	∟	,	<	L	\	l	|	î	£	¼	╝	╠	▄	∞	ⁿ
13	-D	♪	↔	-	=	M	]	m	}	ì	¥	¡	╜	═	▌	φ	²
14	-E	♫	▲	.	>	N	^	n	~	Ä	₧	«	╛	╬	▐	ε	■
15	-F	☼	▼	/	?	O	_	o	⌂	Å	ƒ	»	┐	╧	▀	∩	

Character sets

You can use combinations of keys to produce single characters because the computer differentiates between characters by giving them each a numerical value: each character has a numerical value to represent it, even though it may have been input/produced using more than one key.

The internal code is always written either as a Decimal number or as a Hexadecimal number (numbers to the base 16)

ASCII - American Standard Code for Information Interchange

ASCII is the name for a common character set that has been used for many computers.

The numerical values assigned to the 128 characters in the first half of each code page (see Section 5.1) are the same as those in the ASCII character set.. The full 256 characters in code page 437 is sometimes referred to as the 8-bit ASCII table.

ASCII Table

DEC			16	32	48	64	80	96	112
	Hex Digits	-0	-1	-2	-3	-4	-5	-6	-7
0	-0	NUL	DLE		0	@	P	`	p
1	-1	SOH	DC1	!	1	A	Q	a	q
2	-2	STX	DC2	"	2	B	R	b	r
3	-3	ETX	DC3	#	3	C	S	c	s
4	-4	EOT	DC4	$	4	D	T	d	t
5	-5	ENQ	NAK	%	5	E	U	e	u
6	-6	ACK	SYN	&	6	F	V	f	v
7	-7	BEL	ETB	'	7	G	W	g	w
8	-8	BS	CAN	(	8	H	X	h	x
9	-9	HT	EM	)	9	I	Y	i	y
10	-A	LF	SUB	*	:	J	Z	j	z
11	-B	VT	ESC	+	;	K	[	k	{
12	-C	FF	FS	,	<	L	\	l	|
13	-D	CR	GS	-	=	M	]	m	}
14	-E	SO	RS	.	>	N	^	n	~
15	-F	SI	US	/	?	O	_	o	⌂

5.1 Code Pages

Introduction

If you want to use more than one language, or if you want to use the symbols of a foreign country, to input or output data on your PC you will most probably need to use a different character set for each language or country.

A character set is a group of 256 characters, which is specific to a given country or language, for example, the German character set is the particular set of letters, numbers and symbols used by the Germans; the UK character set is the set used in the United Kingdom, and so on.

A code page is a table that contains the definition of one or more character sets. The definitions are stored as numeric values which the computer translates into letters, numbers and symbols you can recognise on the screen or from the printer.

The process of changing character sets is called code page switching.

Code page switching allows you to:

- use the most suitable set of characters for your own country
- switch your machine to using a different character set on a temporary basis, and back to the default setting again, quickly and easily.

Previously, the IBM PC offered a choice of 256, apparently different, characters per ROM: 256 being the fundamental limit to the number of individual characters in a set, for an 8 bit computer.

The 256 characters chosen, did not include all the variations of alphabets, currencies and other national, linguistic symbols, of European countries. Therefore, European users who needed to use symbols of more than one country, had to make hardware adjustments to access different ROMs, or reset the DIP switches each time they wanted to read or print, a character or symbol which was not one of the 256 characters resident in their ROM code page.

This was very inconvenient for a company in Portugal, for example, which works on an international scale and needs to produce and read documents from Italy, France and the United States, on a daily basis.

Code page switching

Now, however, to accomodate European users, there are five different sets of characters with MS DOS. All five sets are available to your PC. One set is already installed to use as a default setting. It contains those characters most useful to the United Kingdom. There are then four more sets of characters available, which you can switch to, using MS DOS commands.

The five character sets are as follows:

437 The set of characters used in English speaking countries in virtually all PCs, before options existed allowing code page switching.

850 A character set which aims to suit all countries, that is, a multilingual set.

860 Contains certain special characters for Portugal

863 Contains certain special characters for French-Canada

865 Contains certain special characters for Norway/Denmark

Note The numerical value of the code page number is insignificant.

The characters within each of these code pages are shown on the following pages.

DEC	16	32	48	64	80	96	112	128	144	160	176	192	208	224	240
Hex Digits	-0	-1	-2	-3	-4	-5	-6	-7	-8	-9	-A	-B	-C	-D	-E	-F
0 -0		▼		0	@	P	`	p	Ç	É	á	▒	└	╨	α	≡
1 -1	☺	▲	!	1	A	Q	a	q	ü	æ	í	▓	┴	╤	β	±
2 -2	☻	↕	"	2	B	R	b	r	é	Æ	ó	═	┬	╥	Γ	≥
3 -3	♥	‼	#	3	C	S	c	s	â	ô	ú	─	├	╙	π	≤
4 -4	♦	¶	$	4	D	T	d	t	ä	ö	ñ	┼	─	╘	Σ	⌠
5 -5	♣	§	%	5	E	U	e	u	à	ò	Ñ	╞	┼	╒	σ	⌡
6 -6	♠	▬	&	6	F	V	f	v	å	ù	ª	╟	╞	╓	µ	÷
7 -7	•	↨	'	7	G	W	g	w	ç	ù	º	╢	╟	╫	τ	≈
8 -8	◘	↑	(	8	H	X	h	x	è	ÿ	¿	╡	╚	╪	Φ	°
9 -9	○	↓	)	9	I	Y	i	y	ë	Ö	⌐	╣	╔	┘	Θ	∙
10 -A	◙	→	*	:	J	Z	j	z	è	Ü	¬	║	╩	┌	Ω	·
11 -B	♂	←	+	;	K	[	k	{	ï	¢	½	╗	╦	▌	δ	√
12 -C	♀	∟	,	<	L	\	l	|	î	£	¼	╝	╠	▄	∞	ⁿ
13 -D	♪	↔	-	=	M	]	m	}	ì	¥	¡	╜	═	▀	φ	²
14 -E	♫	▲	.	>	N	^	n	~	Ä	Pt	«	╛	╬	▀	ε	■
15 -F	☼	▼	/	?	O	_	o	⌂	Å	ƒ	»	┐	╧	▀	∩	

Code Page 850 - Multilingual

DEC	16	32	48	64	80	96	112	128	144	160	176	192	208	224	240
Hex Digits	-0	-1	-2	-3	-4	-5	-6	-7	-8	-9	-A	-B	-C	-D	-E	-F
0	-0	▼		0	@	P	`	p	Ç	É	á	▒	└	ð	Ó	-
1	-1	▲	!	1	A	Q	a	q	ü	æ	í	▓	┴	Ð	ß	±
2	-2	↕	"	2	B	R	b	r	é	Æ	ó	┃	┬	Ê	Ô	=
3	-3	♥	‼	#	3	C	S	c	s	â	ô	ú	─	├	Ë	Ò	¾
4	-4	♦	¶	$	4	D	T	d	t	ä	ö	ñ	┤	─	È	õ	¶
5	-5	♣	§	%	5	E	U	e	u	à	ò	Ñ	Á	┼	ı	Õ	§
6	-6	♠	▬	&	6	F	V	f	v	å	û	ª	Â	ã	Í	µ	÷
7	-7	•	↨	'	7	G	W	g	w	ç	ù	º	À	Ã	Î	þ	¸
8	-8	◘	↑	(	8	H	X	h	x	ê	ÿ	¿	©	╚	Ï	Þ	°
9	-9	○	↓	)	9	I	Y	i	y	ë	Ö	®	╝	╔	┘	Ú	¨
10	-A	◙	→	*	:	J	Z	j	z	è	Ü	¬	═	╩	┌	Û	•
11	-B	♂	←	+	;	K	[	k	{	ï	ø	½	╣	╦	▀	Ù	¹
12	-C	♀	∟	,	<	L	\	l	|	î	£	¼	╠	╬	▄	ý	³
13	-D	♪	↔	-	=	M	]	m	}	ì	Ø	¡	¢	═	╭	¦	Ý	²
14	-E	♫	▲	.	>	N	^	n	~	Ä	×	«	¥	╬	╮	Ì	¯	■
15	-F	☼	▼	/	?	O	_	o	⌂	Å	ƒ	»	┐	¤	▌	▀	´	

DEC	16	32	48	64	80	96	112	128	144	160	176	192	208	224	240
Hex Digits	-0	-1	-2	-3	-4	-5	-6	-7	-8	-9	-A	-B	-C	-D	-E	-F
0		▶		0	@	P	`	p	Ç	É	á	▒	└	╨	α	≡
1	☺	◀	!	1	A	Q	a	q	ü	À	í	▓	┴	╤	β	±
2	☻	↕	"	2	B	R	b	r	é	È	ó	┤	┬	╥	Γ	≥
3	♥	‼	#	3	C	S	c	s	â	ô	ú	┤	├	╙	π	≤
4	♦	¶	$	4	D	T	d	t	ä	ò	ñ	┐	─	╘	Σ	⌠
5	♣	§	%	5	E	U	e	u	à	ò	Ñ	┼	┼	╒	σ	⌡
6	♠	▬	&	6	F	V	f	v	Á	Ú	ª	╞	╞	╓	µ	÷
7	•	↨	'	7	G	W	g	w	ç	ù	º	╟	╟	╫	τ	≈
8	◘	↑	(	8	H	X	h	x	ê	Ì	¿	╚	╚	╪	Φ	°
9	○	↓	)	9	I	Y	i	y	Ê	Ò	⌐	╔	╔	┘	Θ	∙
A	◙	→	*	:	J	Z	j	z	è	Ü	¬	╩	╩	┌	Ω	·
B	♂	←	+	;	K	[	k	{	ï	¢	½	╦	╦	▌	δ	√
C	♀	∟	,	<	L	\	l	|	Ô	£	¼	╠	╠	▐	∞	ⁿ
D	♪	↔	-	=	M	]	m	}	Ì	Ù	¡	═	═	▀	φ	²
E	♫	▲	.	>	N	^	n	~	Ä	Pt	«	╬	╬	α	ε	■
F	☼	▼	/	?	O	_	o	⌂	Å	Ó	»	╧	╧	ß	∩	

Code Page 863 - French Canadian

DEC	Hex Digits	-0	-1	-2	-3	-4	-5	-6	-7	-8	-9	-A	-B	-C	-D	-E	-F
0	-0		▶		0	@	P	`	p	Ç	É	¡	▒	└	╨	α	≡
16	-1	☺	◀	!	1	A	Q	a	q	ü	È	¬	▓	┴	╤	β	±
32	-2	☻	↕	"	2	B	R	b	r	é	Ê	ó	╫	┬	╥	Γ	≥
48	-3	♥	‼	#	3	C	S	c	s	â	ô	ú	│	├	╙	π	≤
64	-4	♦	¶	$	4	D	T	d	t	Â	Ë	ë	┤	─	╘	Σ	⌠
80	-5	♣	§	%	5	E	U	e	u	à	Ï	¨	╡	┼	╒	σ	⌡
96	-6	♠	▬	&	6	F	V	f	v	¶	û	³	╢	╞	╓	μ	÷
112	-7	•	↨	'	7	G	W	g	w	ç	ù	-	╖	╟	╫	τ	≈
128	-8	◘	↑	(	8	H	X	h	x	ê	¤	Î	╕	╚	╪	Φ	°
144	-9	○	↓	)	9	I	Y	i	y	ë	Ô	⌐	╣	╔	┘	Θ	∙
160	-A	◙	→	*	:	J	Z	j	z	è	Ü	¬	║	╩	┌	Ω	·
176	-B	♂	←	+	;	K	[	k	{	ï	¢	½	╗	╦	█	δ	√
192	-C	♀	∟	,	<	L	\	l	|	î	£	¼	╝	╠	∞	ⁿ
208	-D	♪	↔	-	=	M	]	m	}	ì	¤	¾	╜	═	▌	φ	²
224	-E	♫	▲	.	>	N	^	n	~	À	Ù	«	╛	╬	▄	ε	■
240	-F	☼	▼	/	?	O	_	o	⌂	§	ƒ	»	┐	╧	▀	∩	

DEC	16	32	48	64	80	96	112	128	144	160	176	192	208	224	240
Hex Digits	-0	-1	-2	-3	-4	-5	-6	-7	-8	-9	-A	-B	-C	-D	-E	-F
0	-0	▶		0	@	P	`	p	Ç	É	á	▒	└	╨	α	≡
1	-1	☺	◀	!	1	A	Q	a	q	ü	æ	í	▓	┴	╤	β	±
2	-2	☻	↕	"	2	B	R	b	r	é	Æ	ó	▒	┬	╥	Γ	≥
3	-3	♥	!!	#	3	C	S	c	s	â	ô	ú	▒	├	╙	π	≤
4	-4	♦	¶	$	4	D	T	d	t	ä	ö	ñ	▒	─	╘	Σ	⌠
5	-5	♣	§	%	5	E	U	e	u	à	ò	Ñ	▒	┼	╒	σ	⌡
6	-6	♠	▬	&	6	F	V	f	v	å	û	ª	▒	╞	╓	µ	÷
7	-7	•	↕	'	7	G	W	g	w	ç	ù	º	▒	╟	╫	τ	≈
8	-8	◘	↑	(	8	H	X	h	x	ê	ÿ	¿	▒	╚	╪	Φ	°
9	-9	○	↓	)	9	I	Y	i	y	ë	Ö	⌐	▒	╔	┘	Θ	∙
10	-A	◙	→	*	:	J	Z	j	z	è	Ü	¬	▒	╩	┌	Ω	·
11	-B	♂	←	+	;	K	[	k	{	ï	ø	½	▒	╦	▒	δ	√
12	-C	♀	∟	,	<	L	\	l	|	î	£	¼	▒	╠	∞	n
13	-D	♪	↔	-	=	M	]	m	}	ì	Ø	¡	▒	═	▒	ø	²
14	-E	♫	▲	.	>	N	^	n	~	Ä	Pt	«	▒	╬	▒	ε	■
15	-F	☼	▼	/	?	O	_	o	⌂	Å	ƒ	»	▒	╧	▒	∩	

Using Code Page Switching

Firstly, if you want to print out documents, check that you have either an IBM Proprinter model 4201, or an IBM Quietwriter III printer, model 5202. Code page switching will only be effective in printing if you are using a supporting printer.

Installing code page switching

Your PC already has a single code page installed, which is the United States code page (437). This is the default code page as it is the one which is most appropriate for use in the United Kingdom.

To install a secondary code page, and the facility which allows you to switch from using the default code page (437) to the secondary code page, you will need to edit the CONFIG.SYS and AUTOEXEC.BAT files.

When you install more than one code page, the second code page is always the multilingual code page (850).

1. Edit the file CONFIG.SYS using the command:

   RPED CONFIG.SYS

2. When the current contents of the file are shown on the screen add the following line below the last line:

   DEVICE=DISPLAY.SYS CON:=(EGA,437,1)

3. Press **[Esc]** to save this change in the file.

4. When RPED's main menu appears press **[F1]** to select the option "Edit Existing File" and enter the name AUTOEXEC.BAT when prompted.

5. When the current contents of the file are shown on the screen, press **[Alt]** and **[M]** together to insert a new line at the top of the file. On the blank line that appears type:

   NLSFUNC

6. Press **[Alt]** and **[M]** together to insert another new line and on that line type:

   MODE CON: CP PREPARE=((850) EGA.CPI)

7. Press [Esc] to save these changes in the file.

8. Reset the machine by holding down the **[Ctrl]** and **[Alt]** keys and pressing **[Del]**.

If you intend to use code page switching on a printer install PRINTER.SYS in the CONFIG.SYS file, too. See Part 2, Chapter 8.

Switching to the secondary code page

You will probably find that the default code page is adequate for your day to day use, but if you wish to read or print documents which contain European characters that are not included in your default code page, switch to the multilingual code page.

The multilingual code page contains a set of the most commonly used European characters and symbols.

Having installed the secondary code page and the facility to switch code pages as described above, to switch from the default code page to the secondary code page you will need to be in the directory which contains the CONFIG.SYS file and the AUTOEXEC.BAT file. At the prompt type:

CHCP 850

To return to your default code page type:

CHCP *xxx*

where *xxx* is the code page most appropriate to your country, e.g. the U.K. code page is number 437.

Note The default code page will be reinstalled when you reset your computer.

Changing the default code page

You may also want to use a code page other than the default and multilingual code pages. If, for example, you are in the USA using the USA code page, and you receive a document prepared on a Norway/Denmark system, you would see the symbol for **yen** and the symbol for **cent** on your screen where there should be the characters, **O slash,** and **o slash.**

To see the correct characters for your document, that is the Norway/Denmark (O slash, o slash) you need to switch to the Norway/Denmark code page, **865.**

To use code page 865 you need to change the default from 437 to code page 865.

You cannot change the second page (850) for any other code page.

In order to change the first code page to 865 follow these steps:

1. Edit the file CONFIG.SYS using the command:

   RPED CONFIG.SYS

2. Edit the line which says:

   COUNTRY=044,437

   so that it reads:

   COUNTRY=047,865

   This instructs MS-DOS to use Norwegian country information (i.e. date and time formats, currency symbols, etc.) and also specifies that code page 865, the Norwegian character set, should be the source for the country-specific symbols.

3. Edit the line which says:

   DEVICE=DISPLAY.SYS CON:=(EGA,437,1)

   so that it reads:

   DEVICE=DISPLAY.SYS CON:=(EGA,437,2)

   This instructs the DISPLAY.SYS driver to reserve room for a second code page (both 865 and 850 will be available).

4. Press the **[Esc]** key to save these changes.

5. When RPED's main menu appears press **[F1]** to select the option "Edit Existing File" and enter the name AUTOEXEC.BAT when prompted.

6. Edit the line which says:

   MODE CON: CP PREPARE=((850) EGA.CPI)

   so that it reads:

   MODE CON: CP PREPARE=((850,865) EGA.CPI)

7. Edit the line which says:

   KEYB UK,437

   so that it reads:

   KEYB NO,865

   In order to use the Norwegian code page you must instruct MS-DOS to use the keyboard driver for a country that has 865 as its default code page.

8. Press the **[Esc]** key to save these changes.

9. Reset the PC holding down the **[Ctrl]** and **[Alt]** keys and pressing **[Del]**.

Commands

The MS DOS commands which support code page switching are:

NLSFUNC Loads the file containing country-specific information.

CHCP Displays or changes the current code page for the system and all prepared devices

SELECT Installs MS DOS on a new floppy disk with the selected country-specific information and keyboard code.

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 use this command to select an alternative definition file (other than the default KEYBOARD.SYS file) if alternatives exist.

MODE

- Prepares a code page for a device
- Selects a code page for a device
- Displays the code pages prepared and selected for a device
- Refreshes code pages that were lost due to hardware error

GRAFTABL Displays an extended character set when using display adaptors in graphics mode. A table of data is loaded into memory which defines the additional ASCII characters, 128 to 255.

CONFIG.SYS commands

COUNTRY Identifies the country in which you work or live. This command also defines the appropriate country specific conventions such as date and time formats, and sorting sequence for the character set.

DEVICE Installs device drivers in the system, including two device drivers that can be installed, which 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

The following commands will use country-specific date and time conventions, relevant to the code pages you choose:

DATE
BACKUP
RESTORE
TIME

See Part 1, Section 3.4 for details of country dependant information - e.g. date and time formats, currency symbols, etc.

5.2 Technical Information - Keyboard

Keycodes

The 8-bit keyboard data is capable of 128 'make' codes correspondingly 128 'break' codes. For any key which is pressed, the 'make' keycode produced is in the range of 0 - 127 (decimal). When a key is released, the 'break' keycode produced is the same as the 'make' keycode except that the MS bit is set so that the value is in the range of 128 - 511 (decimal). A number of keys (such as the dedicated keypad keys) send an extended keycode sequence which identify them uniquely as dedicated keys rather than the numeric keypad set. The typical extended keycode sequence consists of an extended keycode identifier, 224 (0E0h), followed by the keycode.

After a key is pressed and the keycode has been sent to the main board electronics, if no new keys are pressed and the key has remained pressed for more that one half second, then the keyboard microcontroller re-sends the keycode every 83 milliseconds provided that the main board indicates by an Acknowledge sequence that is ready to accept a new keycode. The Pause/Break key does not repeat.

The keycode 0AAh is sent after a reset to indicate successful completion of power-up tests.

Keyboard Interrupt.

The ROS Keyboard hardware interrupt reads a key code from the keyboard interface, translates it into a 16-bit key token using an internal translation table and puts this token into the key token buffer. If the buffer is full the key token is discarded and a bleep is output on the speaker. The key tokenization for the most part consists of the high byte being the key number and the lower byte being the ASCII for the keycap. Those keys for which there is no ASCII equivalent the token consists of a unique high byte value with the lower byte cleared. In the case of the new 101/102 key keyboard there are quite a few extra keys. There are extra ALT and CTRL keys as well as a dedicated cursor keypad. These keys basically return the exact same keycode as the originals except that preceding the keycode an extra one is sent to signify that the key pressed is one of the new keys.

The table on the following pages shows the 16 bit keycodes produced by the various keys on the keyboard. This information should be useful to programmers using the ROS keyboard interrupt and should also be useful to anyone using either the **/L** or **/R** parameters of the MOUSE command. The left and right mouse buttons may be set to generate the equivalent of any of the keys by including the appropriates keys keycode value with the **/L** or **/R** parameter. For example:

MOUSE /L0F09 /R1C0D

would set the left button to produce TAB and the right button to produce Carriage Return.

Key Code	Key Name	Normal	ALT	CTRL	SHIFT	Num Lock
01	ESC	011B	01F0	011B	011B	N/A
02	1 and !	0231	7800	Ignored	0221	N/A
03	2 and @	0332	7900	0300	0340	N/A
04	3 and #	0433	7A00	Ignored	0423	N/A
05	4 and $	0534	7B00	Ignored	0524	N/A
06	5 and %	0635	7C00	Ignored	0625	N/A
07	6 and ^	0736	7D00	071E	075E	N/A
08	7 and &	0837	7E00	Ignored	0826	N/A
09	8 and *	0938	7F00	Ignored	092A	N/A
0A	9 and (	0A39	8000	Ignored	0A28	N/A
0B	0 and)	0B30	8100	Ignored	0B29	N/A
0C	- and _	0C2D	8200	0C1F	0C5F	N/A
0D	= and +	0D3D	8300	Ignored	0D2B	N/A
0E	←DEL	0E08	0EF0	0E7F	0E08	N/A
0F	TAB	0F09	A500	9400	0F00	N/A
10	Q	1071	1000	1011	1051	N/A
11	W	1177	1100	1117	1157	N/A
12	E	1265	1200	1205	1245	N/A
13	R	1372	1300	1312	1352	N/A
14	T	1474	1400	1414	1454	N/A
15	Y	1579	1500	1519	1559	N/A
16	U	1675	1600	1615	1655	N/A
17	I	1769	17000	1709	1749	N/A
18	O	186F	1800	180F	184F	N/A
19	P	1970	1900	1910	1950	N/A
1A	[and {	1A5B	1AF0	1A1B	1A7B	N/A
1B	] and }	1B5D	1BF0	1B1D	1B7D	N/A
1C	CR	1C0D	1CF0	1C0A	1C0D	N/A
1D	Ctrl	Ignored	Ignored	Ignored	Ignored	N/A
1E	A	1E61	1E00	1E01	1E41	N/A
1F	S	1F73	1F00	1F13	1F53	N/A
20	D	2064	2000	2004	2044	N/A
21	F	2166	2100	2106	2146	N/A
22	G	2267	2200	2207	2247	N/A
23	H	2368	2300	2308	2348	N/A
24	J	246A	2400	240A	244A	N/A
25	K	256B	2500	250B	254B	N/A
26	L	266C	2600	260C	264C	N/A
27	; and :	273B	27F0	Ignored	273A	N/A
28	' and "	2827	28F0	Ignored	2822	N/A
29	# and ~	2960	29F0	Ignored	297E	N/A
2A	Left SHIFT	Ignored	Ignored	Ignored	Ignored	N/A
2B	\ and |	2B5C	2BF0	2B1C	2B7C	N/A
2C	Z	2C7A	2C00	2C1A	2C5A	N/A

Key Code	Key Name	Normal	ALT	CTRL	SHIFT	Num Lock
2D	X	2D78	2D00	2D18	2D58	N/A
2E	C	2E63	2E00	2E03	2E43	N/A
2F	V	2F76	2F00	2F16	2F56	N/A
30	B	3062	3000	3002	3042	N/A
31	N	316E	3100	310E	314E	N/A
32	M	326D	3200	320D	324D	N/A
33	, and <	332C	33F0	Ignored	333C	N/A
34	. and >	342E	34F0	Ignored	343E	N/A
35	/ and ?	352F	35F0	Ignored	353F	N/A
36	Right SHIFT	Ignored	Ignored	Ignored	Ignored	N/A
37	*	372A	37F0	9600	372A	N/A
38	Alt	Ignored	Ignored	Ignored	Ignored	N/A
39	SPACE	3920	3920	3920	3920	N/A
3A	Caps Lock	Ignored	Ignored	Ignored	Ignored	N/A
3B	F1	3B00	6800	5E00	5400	N/A
3C	F2	3C00	6900	5F00	5500	N/A
3D	F3	3D00	6A00	6000	5600	N/A
3E	F4	3E00	6B00	6100	5700	N/A
3F	F5	3F00	6C00	6200	5800	N/A
40	F6	4000	6D00	6300	5900	N/A
41	F7	4100	6E00	6400	5A00	N/A
42	F8	4200	6F00	6500	5B00	N/A
43	F9	4300	7000	6600	5C00	N/A
44	F10	4400	7100	6700	5D00	N/A
45	Num Lock	Ignored	Ignored	PAUSE	Ignored	N/A
46	Scroll Lock	Ignored	Ignored	Ignored	Ignored	N/A
47	Key Pad 7	4700	Ignored	7700	NOTE 1	4737
48	Key Pad 8	4800	Ignored	8D00	NOTE 1	4838
49	Key Pad 9	4900	Ignored	8400	NOTE 1	4939
4A	Key Pad -	4A2D	Ignored	8E00	NOTE 1	4A2D
4B	Key Pad 4	4B00	Ignored	7300	NOTE 1	4B34
4C	Key Pad 5	Ignored	Ignored	8F00	NOTE 1	4C35
4D	Key Pad 6	4D00	Ignored	7400	NOTE 1	4D36
4E	Key Pad +	4E2B	Ignored	9000	NOTE 1	4E2B
4F	Key Pad 1	4F00	Ignored	7500	NOTE 1	4F31
50	Key Pad 2	5000	Ignored	9100	NOTE 1	5032
51	Key Pad 3	5100	Ignored	7600	NOTE 1	5133
52	Key Pad 0	5200	Ignored	9200	NOTE 1	5230
53	Key Pad .	5300	Ignored	9300	NOTE 1	532E
54	Alt PrtScr	Ignored	Sys Req	Ignored	Ignored	N/A
55	UNDEFINED	Ignored	Ignored	Ignored	Ignored	Ignored
+56	\ and |	565C	Ignored	Ignored	567C	N/A
57	F11	8500	8B00	8900	8700	N/A
58	F12	8600	8C00	8A00	8800	N/A
59 - 7F	UNDEFINED	Ignored	Ignored	Ignored	Ignored	Ignored

NOTE 1: The numeric keypad will produce a different token depending on the states of [NUMLOCK] and the [SHIFT] keys. With [NUMLOCK] on and [SHIFT] down then Num Lock is cancelled and the **Normal** column tokens are produced. With [NUMLOCK] off and [SHIFT] down the tokens in the **Num Lock** column are produced.

Keycode 56 (marked with a '+' above) is handled by the ROM operating system but cannot be produced by the US, 101 key version of the keyboard.

The following table shows the extra keys and their keycodes:

Key Code	Key Name	Normal	ALT	CTRL	SHIFT	Num Lock
E0,1C	Enter	E00D	A600	E00A	E00D	N/A
E0,1D	Right CTRL	Ignored	Ignored	Ignored	Ignored	Ignored
E0,35	Key Pad /	E02F	A400	9500	E02F	N/A
E0,38	Right ALT	Ignored	Ignored	Ignored	Ignored	N/A
E0,46	Prt Scrn	Ignored	Ignored	Ignored	Ignored	N/A
E0,47	Home	47E0	9700	77E0	47E0	N/A
E0,48	↑	48E0	9800	8DE0	48E0	N/A
E0,49	Pg Up	49E0	9900	77E0	49E0	N/A
E0,4B	←	4BE0	9B00	73E0	4BE0	N/A
E0,4D	→	4DE0	9D00	74E0	4DE0	N/A
E0,4F	End	4FE0	9F00	75E0	4FE0	N/A
E0,50	↓	50E0	A000	91E0	50E0	N /A
E0,51	Pg Up	51E0	A100	76E0	51E0	N/A
E0,52	Ins	52E0	A200	92E0	52E0	N/A
E0,53	Del	53E0	A300	93E0	53E0	N/A

The keys from Prt Scrn up to Del, above, are the dedicated cursor keys. They produce different sequences of keycodes depending on the state of the [SHIFT] and [NUMLOCK] keys.

In the base and [SHIFT] + [NUMLOCK] state the key sequence consists of the keycode expected from the corresponding key on the numeric keypad (with the [NUMLOCK] off) preceded by the extra key keycode (ie 0E0h).

With only [NUMLOCK] on the key sequence consists of the base sequence preceded by 0E0h, 2Ah so there are now four keycodes in the sequence.

The last variation is if the [SHIFT] key only is down. In this case the sequence is as shown in the base case preceded by 0E0h, 0AAh and again there are four keycodes in the sequence.

6. THE MOUSE

Your PC is supplied with a mouse. The function of the mouse is to control the movements of the cursor on the screen, by moving the mouse across the surface of the desk. Graphics-based software also uses a pointer which the mouse controls in the same way.

6.1 Using the mouse

The mouse is extremely useful with graphics-based programs such as Windows for example, because the mouse can move the cursor and the pointer over large areas of the screen far quicker than cursor keys.

In graphics-based programs, used for CAD and desktop publishing for example, the pointer is used to select menu options and various facilities all around the screen.

In text-based programs, such as a spreadsheets and word processors, for example, the mouse can be used to perform the same function as the cursor keys.

Loading the mouse software

It is usually necessary to use the MOUSE command before software can recognise the presence of the mouse, although you may find that some software, such as Windows for example, automatically recognises the presence of the mouse. In such cases, it is not necessary to issue any commands to make the mouse work.

The MOUSE command and its associated options are described below.

Note that the **/L** and **/R** options, which redefine the functions of the mouse buttons to produce keyboard functions and characters, are most useful when the mouse is used in text-based programs.

MOUSE External command

MOUSE *[/S] [/F] [/Lxxxx] [/Rxxxx] [/n] [/Xxx] [/Yxx]*

Start the mouse working

Your PC mouse functions the same as the Microsoft mouse when you use the MOUSE command or the MOUSE command with the [/n] option.

In text modes, moving the mouse causes much the same effect as pressing the relevant cursor keys. Pressing a mouse button generates a keycode in a similar way that pressing a key on a keyboard would. However, the mouse buttons are not necessarily active while the mouse is being used by an application program.

Form MOUSE *[option[option...]]*

Options

/S Use slow clock
The PC mouse alters the speed of the clock to provide smooth operation. Some other programs also alter the clock. So, to avoid conflict, the PC mouse can be made to run with a slow clock.

NOTE: You can tell there is a conflict when the time is too fast, or too slow, by a factor of three.

/F Use fast clock
This is used to speed the mouse clock up again after a **/S** command has been used. This faster clock speed allows a much smoother operation of the mouse. It is the default speed.

/Lxxxx Set left mouse button
This assigns a key token in hexadecimal to the left-hand mouse button. By default, this is set to the [↵] key. That is, pressing the left mouse button produces the same effect, on the screen as pressing the [↵] key on the keyboard.

/Rxxxx Set right mouse button
This option assigns a key token in hexadecimal to the right-hand mouse button. By default, it is set to 01BH, the [Esc] key. That is, pressing the right button produces the same effect, on the screen, as pressing the [Esc] key on the keyboard.

/n Use serial port
Where *n* is a number between 1 and 4. This causes the MOUSE program to look for a Microsoft mouse on the serial port. If it fails to find a serial mouse it will then try to initialize an Amstrad mouse.

NOTE: The mouse is now activated as a standard serial mouse and that all the convenient functions of an Amstrad mouse are not implemented. After MOUSE has been run you cannot change from a serial mouse to an Amstrad mouse or vice versa.

/X*xx* Set X scaling
Where *xx* is a two digit hexadecimal number. This alters the sensitivity of the mouse when it is used to simulate the left and right cursor keys. That is, the scaling factor of the mouse is the distance the cursor moves across the screen in relation to the distance the mouse is moved across the table.
0AH is the default setting. Setting values larger than 7FH will cause the key emulation to operate in reverse.

/Y*xx* Set Y scaling
Where *xx* is a two digit hexadecimal number. This alters the sensitivity of the mouse when it is being used to simulate the up and down cursor keys. You can change the distance the cursor moves up and down the screen in relation to the distance the mouse is moved up and down the table. 0AH is the default setting. Setting values larger than 7FH will cause the key emulation to operate in reverse.

Notes

i) All other parameters can be changed by using the MOUSE command again, followed by the relevant parameters. A screen message will appear telling you that the mouse is already working.

ii) With the **/L** and **/R** options the key token values to use may be found from the table of keycodes in the Technical Information at the end of Part 1, Chapter 5. The whole key token need not be used. The last two digits are usually sufficient and are more easily remembered.

iii) If a game does not state explicitly that it uses a mouse, it will probably not recognise the cursor key emulation, since games tend to go directly to the keyboard hardware.

- To activate the PC mouse type:

  MOUSE

  this simplest form of the MOUSE command will set the mouse buttons to [Esc] and [↵]; the mouse clock speed to fast; and the sensitivity to 0AH in all directions. For most applications, this is perfectly adequate.

- To use the mouse with an application which appears to upset the clock, type:

  MOUSE /S

- When you have finished using that application you can use the command:

  MOUSE /F

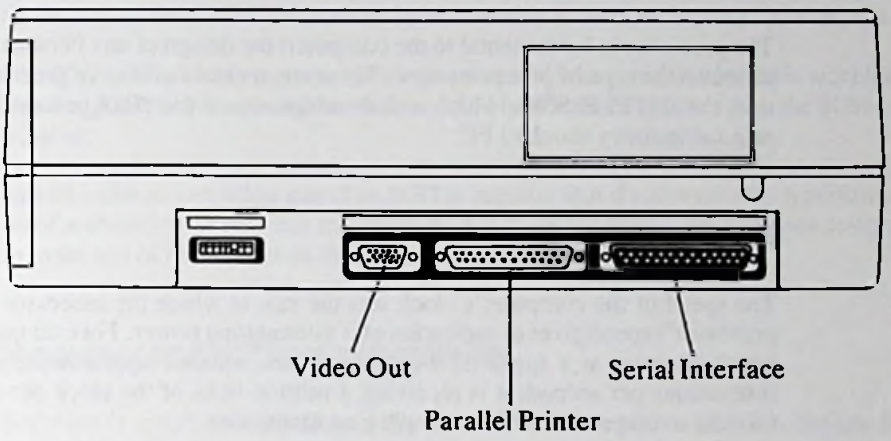
  to return the mouse to its normal, smooth operation.

- If you want the left-hand mouse button to emulate the space bar, and the right-hand button to emulate the [+] key, you can redefine the buttons using the following command:

  MOUSE /L3920 /R4E2B

  The key token values used in this command can be found in the table within the Technical Information section at the end of Part 1, Chapter 5.

7. THE SYSTEM UNIT



At the back of the System Unit you will see

- A parallel port
- A serial interface

On the lefthand side of the System Unit you will see

- Mouse port
- Keyboard Port

On the righthand side of the System Unit you will see

- External floppy disk port

The contents of the System Unit include

- Central Processing Unit (CPU)
- Random Access Memory (RAM)
- Display Adaptor
- Expansion slots for additional external devices
- Power supply
- Disk drives

7.1 The Processor (CPU)

The processor is central to the PC: it is a sophisticated calculating device. It is set apart from a simple calculator by its ability to conditionally process a sequence of instructions. That is, to only perform certain functions on the successful completion of another calculation.

The processor is fundamental to the computer: the design of any Personal Computer is based on the type of processor used. There are several families of processor: Your PC uses the INTEL® 8086, which is a development of the 8088 processor used in the original industry standard PC.

Processor speed

The speed of the computer's clock sets the rate at which the processor operates; the processor's speed gives an indication of it's computing power. For example, a processor which operates at a speed of 4 megahertz will execute approximately one million instructions per second; it is receiving 4 million ticks of the clock per second and it takes an average of 4 ticks to complete an instruction.

The 8086 processor, in your PC, operates at a speed of 8 megahertz which means it executes about 2 million instructions per second.

Processor efficiency

The 8088 and 8086 are both 16 bit processors, but the 8086 has the ability to transfer data more efficiently. This is because it can transfer 16 bit quantities of data; whereas the 8088 can only transfer 8 bit quantities of data, in one instruction.

The combination of this data transfer ability in the 8086, and the speed at which instructions are executed in your PC's processor gives an overall processing speed approximately 2.3 times faster than the industry standard processor which has a clock speed of 4.77 Megahertz.

8087 Maths co-processor

The effective speed of the 8086 can be increased by adding an 8087 mathematics co-processor to the your PC. The 8087 is sometimes referred to as an NDP chip (Numeric Data Processor).

As supplied the your PC does not contain an 8087. However an empty socket exists on the circuit board and your dealer should be able to fit one for you if you require it.

The 8087 is effectively a second processor that will take some of the 8086s workload. It is designed to perform certain mathematical functions far quicker than the 8086 can achieve.

Not all software can make use of an 8087 processor. But if software which performs a lot of arithmetic (for example spreadsheets, CAD/3D programs, etc) has been designed to make use of the 8087 then the program will run faster.

Processor and software compatibility

Software is specified according to the type of processor with which it is designed to work. When buying software for your PC you must ensure that it has been written to work on an 8088 or 8086 processor; the specification should be given on the software's packaging. It would also be wise to check the other requirements of the software against the specification given in Chapter 1.

Because the 8086 and 8087 are highly popular PC processors, there is a vast amount of compatible software available.

Some software would benefit considerably if an 8087 processor is fitted for it to use, though it would still work with an 8086.

At the time of writing, the 80286 and 80386 processors are rapidly gaining popularity and although software that is written for the 8086 will work on these processors, the reverse is not true. That is, you cannot use software on your PC that was written for an 80286 or an 80386 processor.

7. 2 RAM

Introduction

A computer is a calculating device: within its ambit it has

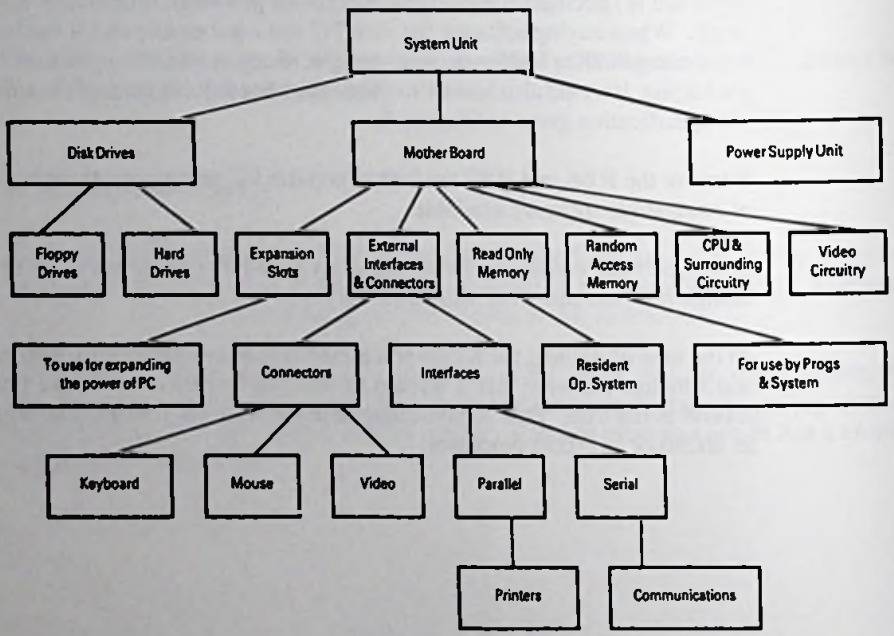
- instructions given to it

- information to be manipulated

It works at high speeds and needs instructions for immediate action and instructions for the action to follow. It also needs to have information close at hand.

Temporary data storage

The area which stores information, while it is being processed, is the RAM (random access memory). The RAM is specially designed so that information stored in the RAM can be accessed and manipulated quickly and easily by the processor.



If the RAM were the only area in which information were stored, the major problems in using the PC frequently would be:

- All the information contained in the RAM would be lost once the computer is switched off.

- Information which is used on more than one occasion would have to be typed in every time it was needed. Also, if you modified the processing instructions and intended to use them more than once, you would have to print out the modified set of instructions, since they too would be lost once the computer was switched off.

- There could never be enough memory space - 640 kilobytes would not be enough.

Permanent data storage

The solution is to provide a permanent, mechanical record and a means of transferring that record in and out of the computer. So that after work is completed you can keep all your data, and reload it whenever you want to. Also, if you run out of memory space, you can start using another record.

Data was first recorded on punch cards, then paper tape and then magnetic tape. The most recent development is that of diskettes which provide a means of loading recorded data into memory. Magnetic tape was re-usable, but diskette recording, as well as being re-usable, is more dense, and therefore more economical. It also has the major advantage that data stored on it can be accessed in a random, rather than sequential, manner. This means that any piece of data is equally accessible. On tape it would be necessary to wind through to the right point while a disk drive can move directly to the required information.

A disk provides a means of extending the memory of the computer: data on the disk is permenantly stored, and is not in use at any given moment, while data in the RAM is there because it is being used - but only temporarily.

Data is loaded from disk into RAM to be used. It can also be loaded from floppy disk to hard disk and then to RAM, and from floppy disk to RAM.

Memory management - swapping data from disk to RAM

The process of loading data from a permanent storage to the RAM will fail when the RAM is full. Many programs will then give you a message telling you that the memory is full.

The RAM's maximum storage capacity is only 640 kilobytes, so when the operating system and application programs have been loaded, there may not be enough space left to hold all the data that needs to be processed. That is, there may not be enough space to hold all the data at once.

However, some programs use a technique of loading only that data which is being processed, then as soon as the processing is finished, this data is put back onto the disk and the next item of data to be processed is loaded into RAM.

This process of moving data, as it is needed, from hard or floppy disk to RAM is called "swapping". Swapping is not the same as simply loading data into RAM, it is a memory management activity: the RAM holds only the data that is currently being processed, then as soon as it is finished, it is put back on the disk, and the next unit of data to be processed is loaded.

Therefore, swapping data back and forth provides a means of using more RAM than the computer actually has.

Using this technique, programs may give the impression they are "multi-tasking", that is, executing two tasks at once. A simple example is that of printing a document whilst editing another document. In fact, the programs are not multi-tasking, one task is being executed at a time: task A is processed first, immediately followed by task B, followed by task A, and so on. The printing process only reads the data from disk as it is required to be sent to the printer.

The swapping facilities within UNIX, OS/2, Desqview and Windows for example, are not apparent to the user: the programs handle data transfer from hard disk to RAM and back again, without instructions from the user.

It takes a lot of time to swap data from RAM to floppy disk and back and it is therefore impractical to attempt to use software that makes use of swapping techniques with floppy disks.

RAM DISKS

If you have a floppy disk drive PC, you may find that it is fundamental to the operation of some programs to store data on a fixed disk temporarily, whilst processing other data.

In which case, you can make the programs treat a portion of the RAM as though it were a fixed disk drive by adding the following line to the CONFIG.SYS file:

DEVICE=RAMDRIVE.SYS *xxx*

where *xxx* defines the amount of memory that should be used to simulate a disk drive. For example:

DEVICE=RAMDRIVE.SYS 138

would apportion 138K of the RAM as memory for the program to treat as though it were a disk drive.

The address of this RAM drive will be the next letter of the alphabet following the last assigned drive letter: if you have a floppy disk drive PC, the drive(s) will already be addressed as A: and B: so this RAM drive will be addressed as C: and so on.

The disadvantage of using some of the RAM for this purpose is that you lose the use of some of the main RAM in its original function.

Remember that the data held in a RAM disk will be lost when the PC is switched off, but the RAM disk space will be present when the PC is switched back on.

Another solution is to add a RAM card to the PC, increasing the RAM size from 640K to anything up to several megabytes. Any extra RAM that you add above the 640K limit will normally be used as a RAM disk; however, there is some software which will address this additional memory directly.

Additional features of the your PC's System Unit

A clock chip is also incorporated into the computer which records the date and time of day, relative to 1st January 1980. It is powered by the batteries in the System Unit. MS-DOS date stamps files and the presence of the clock chip means that it should always use the correct date and time.

However, if you do not set the time and date using the MS-DOS TIME and DATE commands, MS-DOS will stamp the files with the wrong date. Since this facility relies on batteries for power, an incorrect stamp will be given if the batteries need to be replaced.

This feature is not just a luxury; it is useful, for example, when you have several versions of a piece of work, in a number of separate files, and need to know which is the latest version.

You will notice that precautions have been taken to encourage you to set the time and date. This is not a function of the computer; MS-DOS supplied with your PC checks the recorded date and time and if it is not valid it will prompt you to enter the correct settings when it first loads.

7.3 Expansion slots

Expansion slots are used to connect expansion cards to the main mother board.

Expansion cards are sections of electronic circuitry, each on a single piece of printed circuit board. They are used to provide a PC system with extra functions.

Industry standard PC's are designed, normally, to be flexible, and expandable, leaving room for features to be added according to individual requirements.

Some features are necessary for fundamental use of the PC. Those features which are necessary or most popular have been built into the your PC:

- floppy disk controller
- graphics controller (video adaptor)
- mouse adaptor (interface card)
- parallel interface
- serial interface (asynchronous communications adaptor)
- hard disk controller (if you have a hard disk)

There are three additional expansion slots to allow the PC to be upgraded for more specialised requirements, for example, a modem expansion card could be added, or an LAN (Local Area Network) to allow the PC to be incorporated in a network of computers.

Therefore, your PC is equipped with all the necessary features plus the additional features that are most usually required. It also allows for an ongoing process of upgrading, as more advanced needs are realised.

Although the graphics adaptor is a built-in feature, you can still change it if you wish. There is a facility provided to disable the built-in graphics adaptor: adjust DIP switch, number 4, on the back of the system unit, to the ON position (i.e. down).

The serial interface can be disabled too. You may buy a modem card, for example, which has a serial interface built into the card which conflicts with that of the PC (i.e. COM1) in which case disable the PC serial interface by setting DIP switch number 3 to the ON position (i.e. down).

Inserting an expansion card

1. Switch off the PC and any attached equipment.

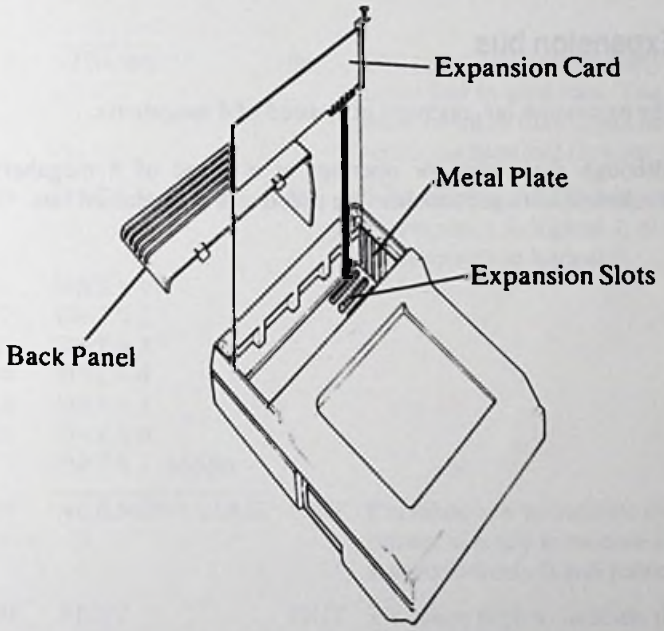
2. Disconnect printers and any other attached equipment from the System Unit. Also disconnect the Display from the System Unit.

3. Lift the Display out of the recess in the top of the system unit and put it to one side.

4. Undo the screw and remove the panel on the upper surface at the back of the System Unit by pressing the two squares on the panel which are marked with arrows and sliding the panel towards the back of the System Unit.

   If this is the first expansion card to be installed in your PC, slide back and remove the panel on the lefthand corner of the System Unit.

   Inside the Unit and on the righthand side you will see 3 expansion slots.



5. Remove the expansion slot screw - turn the screw anticlockwise to undo it. Then remove the metal plate that covers the hole at the right-hand end of the expansion slot. Keep the screw and metal plate.

6. Press the expansion card firmly into the slot: hold the expansion card by its top edge and position it above the slot.

7. Insert the screw on the expansion card into the screw hole of the expansion slot and tighten it by turning the screw clockwise.

8. - Replace the panel on the top of the System Unit and fix it into place with the screw at the centre of the panel. If the expansion card is not to be fitted to an external device the panel at the righthand side of the System Unit may also be replaced.

   - If the expansion card is to be fitted to an external device, connect the appropriate cable to the connector which will now be showing on the righthand side of the System Unit.

9. Replace the Display on the top of the System Unit.

10. Re-attach all cables.

11. Run the install program for the expansion card, if there is one supplied with it.

Expansion bus

The expansion bus operates at a speed of 4 megahertz.

NOTE Although the processor operates at a speed of 8 megahertz, it is automatically synchronized to accomodate the speed of the expansion bus.

7.4 Parallel printer port

The parallel printer connector on the back of the System Unit is a 25-way D-type connector, which provides a standard parallel (Centronics) interface and so can be used to connect any printer that uses a standard parallel interface. This gives you a wide choice of printers and plotters, though we would naturally recommend the Amstrad DMP and LQ range.

If you or your dealer are not sure whether a printer or plotter is suitable, compare the pin-out and timing information given for the printer/plotter with the pin-out and timing information below.

In the following table, **C Pin** is the pin number on the Computers parallel port, **P Pin** is the pin number on the Printers connector and **R Pin** is the Return pin. The **I/O** indication shows the direction of the signal relative to the printer.

C Pin	P Pin	R Pin	Signal name	I/O	Notes
1	1	19	$\overline{\text{STROBE}}$	IN	The signal level is normally high; it is taken low to send data. The pulse width must be more than 0.5uS at the receiving terminal (ie your printer).
2	2	20	DATA 0 (LSB)	IN	8-bit data signal. Pin taken high corresponds to logical 1; pin taken low corresponds to logical 0.
3	3	21	DATA 1		
4	4	22	DATA 2		
5	5	23	DATA 3		
6	6	24	DATA 4		
7	7	25	DATA 5		
8	8	26	DATA 6		
9	9	27	DATA 7 (MSB)		
10	10	28	$\overline{\text{ACKNOWLEDGE}}$	OUT	Pin taken low to indicate that your printer is ready to receive further data. Approximately 0.5uS pulse.
11	11	29	BUSY	OUT	Pin taken high to indicate that your printer cannot receive data.
12	12	30	PE	OUT	Pin taken high to indicate that your printer is out of paper.
13	13	-	SELECT	OUT	Pin taken high to indicate ON-LINE; pin taken low to indicate OFF-LINE.

C Pin	P Pin	R Pin	Signal name	I/O	Notes
14	14	-	AUTOFEED	IN	Pin low means paper to be fed one line after printing.
-	15	-	NC		Not Connected.
-	16	-	0V		Logic GND.
-	17	-	CHASSIS GND		Printer chassis GND.
18	18	-	NC		Not Connected.
19	19	-	GND		Twisted-pair Return signal GND.
:	:	:	:		: : : : : : :
25	25	-	GND		Twisted-pair Return signal GND.
26	26	-	GND		Twisted-pair Return signal GND
:	:	:	:		: : : : : : :
30	30	-	GND		Twisted-pair Return signal GND
16	31	-	INIT	IN	Pin normally high; pin taken low to reset printer controller to its initial state and clear the printer buffer. Pulse width must be greater than 0.5uS at receiving terminal (ie your printer).
15	32	-	ERROR	OUT	Pin taken low to indicate printer out of paper, off-line or in an error state.
-	33	-	GND		Twisted-pair Return signal GND.
-	34	-	NC		Not Connected.
-	35	-	5V		Pulled up to +5V through a 4K7 resistance.
17	36	-	SLCT IN	IN	Data entry to printer only possible when pin low. Pin taken high to indicate printer OFF-LINE.

The "standard" IBM PC to Centronics printer cable should work correctly. This is a cable having a 'male' 25-way D-type connector on one end that will fit into the 'female' 25-way D-type on the back of your PC System Unit. The connector at the other end of the cable may vary but as a general rule it would normally have a 34-way "Amphenol" connector.

The 'male' end of the cable should be plugged into the parallel port on the System Unit and the other end to the equivalent connector on the printer/plotter. If the 'male' plug on the end of the lead has slots for screws, use the screws supplied with the lead to

attach it securely to the Parallel Port. Look at the printer/plotter's own manual to see how it recommends securing the lead to its connector.

If you attach a printer, find out from its manual which character set the printer uses. If it doesn't use the UK 8-bit ASCII character set, see if there is any switch on the printer that can change the character set used by the printer.

It may be necessary to make Windows aware of the type of printer you have attached, especially if you intend to use it to print graphics. See the Windows User Guide for more detail of how to do this.

7.5 Serial port

The serial interface connector on the back of the system unit is a 25-way D-type connector, which provides a standard RS232C interface and so can be used to connect any communications link or printer that uses the standard RS232C serial interface. This gives you a wide choice of modems, communications links and printers.

If you or you dealer are not sure whether a communications link or a printer is suitable, compare the pin-out given for the device with the pin-out details in the following table:

Pin	Description	Signal name
1	Frame Ground	
2	Serial Data Output	TxD
3	Serial Data Input	RxD
4	Request to Send Output	RTS
5	Clear to Send Input	CTS
6	Data Set Ready Input	DSR
7	Signal Ground	GND
8	Data Carrier Detect Input	DCD
9..19	Not used	
20	Data Terminal Ready Output	DTR
21	Not used	
22	Ring Indicator Input	RI
23..25	Not used	

To make connections between the device and your PC, you will need a suitable cable.

The type of cable you need depends on what you want to connect to the Serial Port:

- To connect to a Modem or Desktop computer wired as a modem, you need a '**one-to-one**' RS232 cable.

- To connect to a Serial Printer, a Terminal or a Desktop computer wired as a Terminal you need a '**Null Modem**' RS232 cable.

  One way of checking whether a device is wired as a modem or as a terminal is to look at the type of connector on the device. Conventionally, devices wired as terminals have 'male' connectors and devices wired as modems have 'female' connectors. However, many manufacturers put 'female' connectors on both types of equipment for reasons of electrical safety, so the only sure way of telling is by finding out from the device's manual how Pin 2 is wired:

- If Pin 2 is used to **transmit** (ie Pin 2 is an output pin), then the device is wired as a terminal and you need a **Null Modem** cable.

- If Pin 2 is used to **receive** (ie Pin 2 is an input pin), then the device is wired as a modem and you need a **one-to-one** cable.

  The cable you use should have a 'female' plug on one end to fit the 'male' connector on your PC. The type of plug it should have on the other end depends on whether the device you are connecting to has a 'male' or 'female' connector. Your dealer should be able to help you here.

  The 'female' end of this cable should be plugged into the Serial Port on the back of your PC and the other end into the equivalent connector on the device. If the 'female' plug on the PC end of the lead has screw holes, use the screws supplied with the lead to attach it securely to the Serial Port. Look at the device's own manual to see how it recommends securing the lead to its connector.

  The next step is to set up the Serial Interface so that data is transmitted with the right characteristics and received data is correctly interpreted. Your device's manual should tell you what values you need to set.

  If you plan to use the Serial Interface almost entirely to link your PC with a particular printer, terminal or modem it would be a good idea to include the necessary MODE commands (for details see the description of the MODE command in Part 2) in the AUTOEXEC.BAT file on your MS-DOS disk (or hard disk if you have one). These commands could be added easily using the RPED editor.

  If you are going to use a serial printer with windows it will be necessary to run the CONTROL program to instruct it that the printer is attached to COM1. Details of how to do this may be found in the Windows Users Guide.

  If you add an expansion card providing a COM1 serial port that would conflict with the electronics of the built-in serial port you can disable the built-in interface by switching DIP switch 3 in the bank of DIP switches at the rear of the System Unit.

7.6 Hard Disk

Introduction

A hard disk can store a large number of programs and a large amount of data: a 32 megabyte hard disk has approximately the same storage capacity as forty-five floppy disks.

A hard disk operates ten times faster than floppy disks, and allows immediate access to all your programs, without having to change floppy disks frequently.

It is important to keep the data on your hard disk well organised, with related data and programs grouped together; otherwise your data will be difficult to manage.

Preparation of the hard disk

The version of the PC 2086 which is fitted with a hard disk, is produced with the hard disk already prepared with MS-DOS as the operating system.

Chapter 3, "Installation", describes the actions you must take if, for some reason, your hard disk has not been configured properly.

This section explains how the disk has been configured during manufacture (ie the choices made), the various stages involved in preparing a hard disk, and the reasons why it is configured as it is.

It then goes on to explain alternative configurations; why you may want to reconfigure the disk and how you go about reconfiguring the disk.

Note: If you wish to use any hard disk other than that which was supplied with your PC, you should pay particular attention to the manufacturer's instructions when formatting.

The hard disk configuration as supplied

The hard disk has been configured using MS-DOS. The FDISK command has been used to create a single partition. This means that the disk is prepared for use only with MS-DOS. You cannot use any other operating system, either in place of MS-DOS or as well as MS-DOS.

The hard disk in the PC has 32 megabytes of storage space and the supplied configuration allows MS-DOS to use the whole of the 32 megabytes of storage space as a single drive i.e. drive C:

This configuration is chosen because it is the best setup for most purposes of this PC set-up

The preparatory steps taken to acheive this configuration are:

- HDFORMAT
- FDISK
- SELECT A: C: 044 UK

HDFORMAT is a low-level format that clears the disk of any data, effectively producing a cleared disk. It does this by writing new sector information onto the disk.

FDISK is the MS-DOS command which is used next. It defines the layout of the disk by defining the number and size of the partitions and by setting up the root directory.

A primary DOS partition is created which uses the entire disk for MS-DOS. This single partition is automatically made active.

SELECT A: C: 044 UK is the MS-DOS command used to format the disk. It checks that the hard disk has no bad sections, marking entries in the directory, if necessary, so that bad sectors of the disk cannot be used. Up to 1% bad sectors is acceptable. It installs MS-DOS and puts the CONFIG.SYS and AUTOEXEC.BAT files into the root directory of drive C: These files will contain the necessary commands to set the date and time formats etc. appropriate to the chosen country; **044 UK** is the option for the United Kingdom.

The contents of the CONFIG.SYS and AUTOEXEC.BAT files are described in Part 1, Sections 3.4 and 3.5

Reconfiguring the hard disk

It is possible that the default (supplied) configuration is not the most suitable to your requirements. This section outlines alternative configurations. For detailed instructions on how to reconfigure the hard disk, using MS-DOS, see the FDISK command in Part 2, Chapter 8.

If you decide to reconfigure your hard disk using FDISK, be sure to have copies of any files existing on the hard disk, that you want to keep, as FDISK destroys all existing files.

You will need to reconfigure the hard disk if you wish to use any of the following:

- an operating system other than MS-DOS

- a hard disk with a capacity greater than 32 megabytes

- logical drives

- other operating systems as well as MS-DOS

First, you must use the command HDFORMAT. This is a low-level format that clears the disk of any data, effectively producing a cleared disk. It does this by writing new sector information onto the disk.

If you intend to use an operating system other than MS-DOS, use that operating system to configure the disk.

Otherwise, FDISK is the MS-DOS command which is used next, to define the layout of the disk by defining the number and size of the partitions and by setting up the root directory.

FDISK provides a number of options which are listed below. A series of menus are displayed to help you create and delete partitions. The reason for creating partitions is to separate the hard disk into individual areas to use as separate drives or for use by other operating systems.

Before you use the FDISK command, you should decide whether you want to create a single partition or more than one partition: you will have to create a primary DOS partition first and the menu used to make this selection will also ask you whether you wish to use the maximum size for a DOS partition and make the DOS partition active. If you say yes, you will not then be able to make an extended partition, unless you start the command again.

Using MS-DOS on a disk which is larger than 32 megabytes

The maximum size of a DOS partition is 32 megabytes.

If you have a disk which is larger than 32 megabytes, you should create more than one partition, otherwise MS-DOS will not allow you to use the total disk capacity.

First, you have to create a primary DOS partition, and then an extended DOS partition. You can then go on to create logical drives.

Using logical drives

Logical drives provide a means of breaking up a physical hard disk into distinct areas which may be treated as separate physical units.

This is useful for keeping different types of data apart. If, for example, application programs are kept in one logical drive and data files in another, it would be easy to manipulate collections of data: the contents of a logical drive could be wiped off the disk by reformatting just that drive. But the contents of the whole hard disk would not be lost.

This is therefore, a more powerful division than that provided by subdirectories, which is another way of organising and separating data on a hard disk.

Any size of disk can use logical drives.

You can apportion disk space to the logical drives as appropriate to your requirements. For example, if you have a 40MB hard disk you could apportion the space with one logical drive having 30MB and a second having 10MB. Or 4 logical drives with 10MB each and so on.

First, you should create a primary DOS partition and then an extended DOS partition and then logical drives.

You can create up to 16 logical drives. Given that drives A and B refer to the floppy disk drive; the letter 'C' will be assigned to the first logical drive, then D to the second logical drive, then E and so on.

Using more than one operating system

If for some reason you wish to use more than one operating system it will be supplied with a program equivalent to the MS-DOS FDISK command. Partions on the hard disk reserved by such a program will be described by FDISK as "non DOS partion".

FDISK options

You can use FDISK to:

- Check whether a hard disk is configured or formatted
- Create a primary or single DOS partition
- Create an extended DOS partition
- Create logical drives in extended partitions

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

When you have configured the hard disk, you must format the disk.

You can use

either

the FORMAT command with the /S option.

The FORMAT command checks the hard disk for bad sectors and marks entries in the directory so that these sectors (areas) will not be used.

or you can use the SELECT command which:

- Formats the disk
- Creates both the CONFIG.SYS and AUTOEXEC.BAT files on the new disk
- Copies the contents of the source disk, file by file, to the target disk, installing MS-DOS with the keyboard layout and the date and time format of the country of your choice.

Safety

Your hard disk has been designed to be extremely reliable and to have a long life, but like all mechanical devices, it could break down or be damaged by misuse or accident. There is also a possiblilty that the wrong command could be typed, and data lost. The hard disk can always be repaired or replaced if it goes wrong; what is potentially much more serious is the loss of data that might occur after such a mishap. Ten or twenty megabytes can easily represent years of work, irreplaceable financial records, or invaluable customer records.

This problem has existed ever since the start of computing. A process called 'backing-up', or 'making a back-up' has been evolved by professional computer users. A 'backup' is jargon for a copy made for safety's sake. Backups can be made of the entire hard disk every day (or even more often), or just a file or two every week or fortnight.

In general, backups are made by copying files from the hard disk onto floppy disks, as this does not need any extra equipment and is quite convenient. Often, however,

backups are made with the help of additional tape-based hardware. If you have such equipment, follow the instructions supplied therewith.

Backups can be quite time consuming, and you should weigh up the advantages of security against time taken to create the backups. For example, if you are writing a book using your PC, the loss and subsequent re-typing of a morning's work might be an acceptable risk. Anything more than a day's work lost would probably be unacceptable - in which case, you would want to make a backup daily (i.e. at the end of each day's work).

Every effort has been made to keep the process of making backups simple, but the task, nevertheless, demands a certain amount of discipline.

You will need a number of blank (i.e. formatted) floppy disks.

A hard disk has the capacity of approximately forty-five floppy disks, so at first sight, backing up a thirty-two megabyte hard disk onto forty-five floppy disks seems an awesome undertaking. Since each floppy can take over a minute and a half to write it could seem that backing up would take at least an hour and a half. Fortunately, this is often not the case. The BACKUP program you are supplied with in MS-DOS, has various labour-saving options. You can just copy parts of the hard disk, or files that have been changed since the last backup. Also, you will not need to backup any programs that originally came on floppy disks (as long as you keep them somewhere safe) because you can always reinstall them. But you should keep backups of any information you generate.

Exactly of what, and how frequently, you make backups is therefore up to you. A popular method is to have five disks, one for each working day, onto which that day's work is backed up. Then at the end of the week, everything important is copied onto a master set of floppy disks, and the day by day floppies can be re-used. It is a good idea to have at least two sets of master disks in use as backups, and to alternate between them (i.e. backup onto one set one week, and onto the other the next).

This way, if you have a power cut or accident whilst you are making the backup, and both the hard disk and the floppy are ruined, you can always go back to your last backup. This would not be possible if you had just used one set, as you would have just lost your last backup by copying over it!

On a daily basis, this scheme would require the **/M** option of the MS-DOS BACKUP command, and on a weekly basis, use the **/D** option. See Part 2, chapter 8, for full details of the BACKUP command.

Whatever scheme you choose, it is important to make backing-up a habit that is religiously observed.

Transporting the hard disk

Whenever you move your hard disk PC, take the following precautions:

- Always park the heads first (see section below entitled 'Parking the heads')
- Do not move the unit from a cold environment (e.g. the boot of a car) into a warm environment and attempt to use it immediately - wait until the unit has reached room temperature.
- Take special care when transporting and handling the system unit (which contains the hard disk drive) - it is very sensitive to vibration and will almost certainly be damaged if you drop it.

Parking the heads

Before you move your PC, whether across the office or across town, it is necessary to park the hard disk mechanism's read/write heads. To perform this operation, run the PARK.COM utility, as below. (The PARK.COM utility is provided on a supplied PC system disk, and is installed during the installation process - Chapter 3).

At the **C>** prompt type

PARK

After a few moments, you will see the message

The heads are parked - You may now switch off or press [Esc] to return to DOS.

You may now switch off the PC within ten seconds of seeing this message unless you wish to continue using the hard disk, in which case you should press **[Esc]** and you will see the message

[Esc] : Returning to DOS . . .

If you receive any messages other than those shown above, consult the chapter entitled "Troubleshooting".

7.7 Technical Information

7.7.1 Serial interface

Your PC serial interface is configured as COM1. Display Selector switch 3 can be used to disable the serial interface if an expansion card with a serial interface that must also be COM1 is used.

Programming the Serial Interface

In addition to the four standard serial I/O functions supported by ROS Int 20 (14h), a PS/2 extension has been incorporated into your PC. This extension allows full access to the 8250 setup options and extended baud rate selections. Its calling sequence is as follows:

Serial Int 20 Function 4: Extended Initialize Serial Port

Entry: AH = 4
AL = Break.
0 - no break
1 - break
BH = Parity.
000 - none
001 - odd
010 - even
011 - stick odd
100 - stick even
BL = Stop Bits.
0 - one stop bit
1 - greater than one depending on word length
CH = Word Length.
00 - 5 bits
01 - 6 bits
10 - 7 bits
11 - 8 bits
CL = Baud Rate Code.
000 - 110 baud
001 - 150 baud
010 - 300 baud
011 - 600 baud
100 - 1200 baud
101 - 2400 baud
110 - 4800 baud
111 - 9600 baud
1000 - 19200 baud

DX = Serial Channel Number (0 or 1).

Exit: AX = Serial Status
All flags and other registers preserved.

7.7.2 Hard Disk

The hard disk in your PC uses the latest RLL technology. This allows a greater density of recording on the disk surface. You should only use a certified RLL disk drive mechanism such as is supplied with the controller.

Although an RLL drive has more physical sectors than the older MFM drive, the controller on your PC performs a transparent sector translation so that software appears to be using an MFM drive.

Typical parameters for a WD232R, 30MB drive are:

Physical: 615 tracks, 4 heads, 26 sectors/track
Virtual: 470 tracks, 8 heads, 17 sectors/track

7.7.3 Non Volatile RAM (NVR)

Your PC contains an area of Non-Volatile RAM (NVR). This does not appear directly within the PC's address space but may be accessed by using I/O instructions to address the Real Time Clock (RTC) chip.

The first 40 bytes of the battery backed up RAM within the RTC hardware are for system parameter storage as follows:

Byte(s)	Usage	Default
0 - 9	RTC Time and Date parameters.	--
1	RTC Control Register A.	070h
11	RTC Control Register B.	002h
12	RTC Control Register C.	--
13	RTC Control Register D.	--
14 - 19	Time and Date machine last used	--
20	Century.	
21	Time and Date Checksum.	
22	Floppy Disk Configuration Byte	031h

Bit	Usage
7	Disable Write Precomp.
6	Drive 1 has no changeline
5 - 4	Drive 1 type
3	Drives 0/1 swapped
2	Drive 0 has no changeline
1 - 0	Drive 0 type

Drive types as follows:

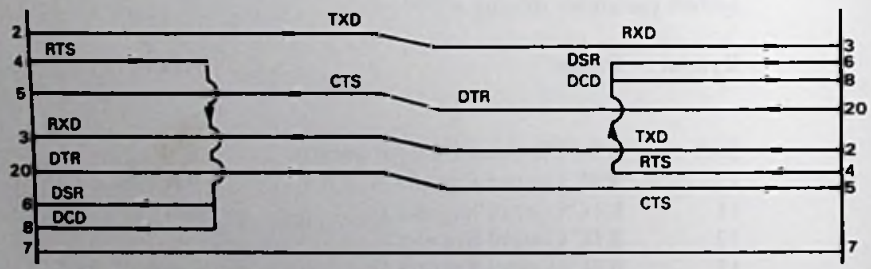
0 0 - 360K Byte
0 1 - 720K Byte
1 0 - 1.44 M Byte
1 1 - Unused (Drive Not installed)

23	Floppy Disk Configuration Byte.	(as above)	031h
24 - 63	Unused	Zeros	

Locations 0 to 13 are RTC hardware registers.

The NVR is checksummed during ROS initialization, and if incorrect then it is initialized with "sensible default values." These consist of date set to 1-1-1980 and floppy disk drives assumed to be 720K Bytes with changeline.

Bytes 22 and 23 are duplicates of each other but perform a different function within the ROS in that the contents of byte 22 is copied to byte 23 during powerup initialization and then byte 23 is used by the floppy disk BIOS to define the active floppy disk parameters. The DEVICE SETFD utility changes byte 22 but byte 23 must not be changed until MS-DOS is reloaded.



Recommended Null-modem cable wiring

8. FLOPPY DISKS

8.1 Buying & using floppy disks

Floppy disks have undergone an enormous amount of technological development. At the same time they have accumulated a good deal of jargon terms.

In simple terms, disks are described according to their storage capacity which is measured in thousands or millions of bytes. The disk capacity is often called the disk size. These measurements provide a means for identifying disks of one generation of computing technology, to another generation.

Your PC encompasses a very wide range of floppy disk technology, which means that you can use two generations of low-density disks on your PC without having to add extra hardware to accomodate the different types.

Therefore, you can use disks which are perhaps considered to be out of date, on a PC which is based on up to date technology.

The four generations of floppy disk technology are:

1st generation - low density 8 inch disks with 160 kilobytes of storage space.

2nd generation - low density 5.25 inch disks with 360 kilobytes of storage space and 3.5 inch disks with 720 kilobytes of storage space.

3rd generation - high density 5.25 inch disks with 1.2 megabytes of storage space

4th generation - high density 3.5 inch disks with 1.4 megabytes of storage space

Your PC can use the 2nd and 4th generation disks.

The functions of floppy disks are to provide the means of transporting software and data from one PC to another. They are used to store progams that you buy; you can then use the disks to load the programs into your PC, or copy the programs from the floppy disks to a hard disk.

One advantage floppy disks have is that they are easier to use than hard disks, initially: hard disks need to undergo a complicated preparation process, involving formatting, configuring with information which will be used whenever the hard disk is used. This involves making decisions about how you intend to use the disk for all your future work. If you should need to change this at a later date, you would have to save all the work held on the disk and re-format and re-configure the disk.

However, processing and using floppy disks is time consuming: formatting, loading and saving of data is much slower using floppy disks.

Although you still need to use floppy disks with hard disk PCs, the majority of your work will be done on the hard disk.

The types of floppy disk you should buy to use on your PC are:

3.5 inch
double sided
soft sectored
135 tracks per inch

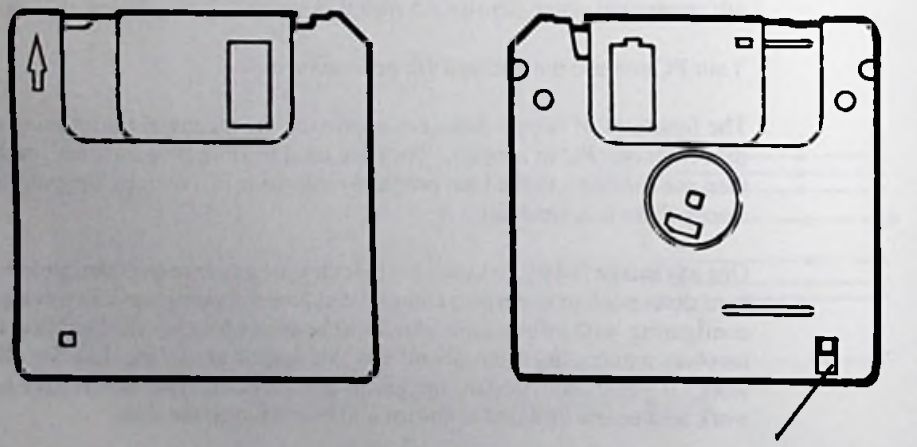
Before you can use a new disk for the first time you must format it. If you are using MS-DOS as the operating system on your PC, see Part 2 for instructions on how to do this. The command you will need to use is the MS-DOS FORMAT command which is listed alphabetically with all the other commands, in Chapter 8.

You should take care of your disks by not putting them near any magnetic sources such as televisions, telephones or hi-fi loudspeakers.

Disks should not be subjected to excessive heat or cold and should not be left in direct sunlight.

Take care not to allow your disks to become dusty, and never touch the magnetic surface beneath the protective metal shutter of the disk.

If you want to write protect the data on the disk, that is, prevent the contents from being accidenatlly wiped off or over written, OPEN the write-protect shutter. This is the small slider at the bottom left corner of the disk.



8.2 External floppy disk drive connector

There is an external floppy disk drive connector on the righthand side of the system unit. An external floppy disk drive could be useful to any one of the three PC models (single drive, dual drive or hard disk).

A single drive PC or a hard disk PC - connecting a second floppy disk drive would make disk handling generally more convenient for you, for example with disk-copying processes, but more importantly, it would give the PC the facility to exchange data from a disk of one size to a disk of a different size. You might also want to add a second floppy disk drive just to increase the amount of storage space available. While running programs the PC could use the disk in one drive to hold the programs while the second drive holds a disk with data.

A dual floppy disk drive PC - connecting a third floppy disk drive would allow you to effectively replace drive B (the righthand drive). Thus providing you with two disk drives of different sizes. This could be useful if, for example, you would like to transfer data from a 5.25 inch floppy disk to a 3.5 inch floppy disk on a PC that has two 3.5 inch disk drives built into the System Unit. It would still be possible to switch back to the built-in drive B if the need arises.

External floppy disk drive sizes

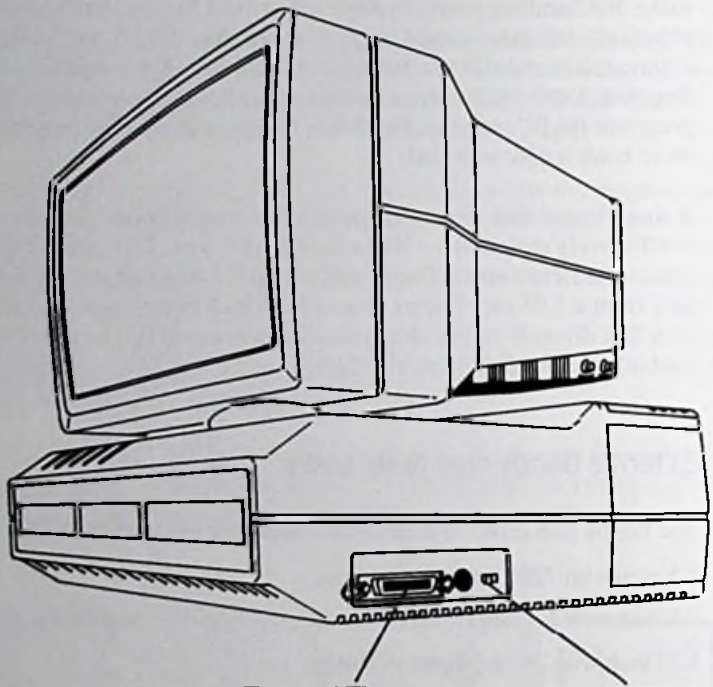
The floppy disk drives that are available are for use with the following disk sizes:

- 3.5 inch with 720 kilobytes of storage
- 3.5 inch with 1.4 megabytes of storage
- 5.25 inch with 360 kilobytes of storage

How to use the external floppy disk drive connector

1. You must make sure the power to the PC system is switched OFF.

2. Then switch the Drive Select switch to EXT. This switch is on the lefthand side of the System Unit, marked INT/EXT. It will indicate to the system that the external drive is the one to recognise as drive B, rather than the original drive B, built into the System Unit.



3. Then plug the ribbon cable from external disk drive to the large connector on the side of the System Unit.

4. Plug the power cable for the external drive into the 4 pin connector on the right-hand side of the System Unit.

5. Once both cables have been connected you may switch the System Unit on.

6. You must make MS-DOS aware of the type of drive that has been added. You do this by using a software command. There are two commands that can be used: the DEVICE command and the DRIVPARM command.

   Note that after you have changed the parameters using either DRIVPARM in the CONFIG.SYS file or the DEVICE command, the changes will not be effective unless

the PC is restarted, using a soft reset by holding down the [Ctrl] and [Alt] keys and pressing [Del] or a power-on reset. This is because MS-DOS sets up the PC according to the configuration instructions it reads when it is loaded. If the configuration instructions change after MS-DOS has been loaded, the new instructions will not be read until MS-DOS is loaded again.

Using the Device command

The DEVICE command is the simplest to use. If you give the command:

DEVICE

it will display various information about the system setup. Towards the bottom of the display you will see the floppy disk configuration parameters set-up that is, the configuration as pre-defined. This shows the size and type of drives that the system believes to be present. It will also tell you whether the disks have the changeline facility.

Changeline is the facility which allows the system to automatically detect that a diskette has been taken out of a drive. While the disk drive can indicate that a disk has been taken out of the drive, the system does not have to waste time checking whether a disk has been taken out or changed.

Usually, 3.5 inch drives do have this facility; 5.25 inch drives do not. The computer needs to know whether a drive has this facility.

Note that it will not matter if you indicate that the drive has not got changeline when in fact it has. But you will have difficulties if you indicate that the drive has got changeline when it has not.

An advantage of using DEVICE rather than DRIVPARM is that a facility called "Drive swapping of A and B" is available with DEVICE.

This facility allows the physical external drive to be drive A and the physical internal drive to be drive B - that is the drive which is built into the system unit can become drive B.

Almost all software will co-operate with this facility. In particular, a soft reset: holding down the **[Ctrl]** and **[Alt]** keys and pressing the **[Del]** key, or a power-on reset will co-operate and therefore will restart the system from the external drive. This is most useful if you want to use some software which loads from a 5.25 inch floppy disk.

In order to use the DEVICE command to make the system aware of the type of external drive that you have added, it would be used in the form:

DEVICE SETFD /D:1 /T:*n [parameter] [parameter..]*

The value of *n* indicates the type of drive that has been added and may take the following values:

n=0	if a 360K 5.52 inch drive has been added
n=1	if a 720K 3.5 inch drive has been added.
n=2	if a 1.4M 3.5 inch drive has been added.

The optional parameters may be used to indicate that changeline is supported or that drive A and B should be swapped. Further details of using the DEVICE command may be found in Part 1, Chapter 3.3

For example, supposing a 1.4M 3.5 inch drive with changeline support has been added and this should be recognised as drive A (the internal drive therefore becoming drive B). the following command would achieve this:

DEVICE /D:1 /T:2 +C +S

8.3 Technical Information - Floppy Disk Controller

Your PC contains a floppy disk controller capable of operating with a number of different types of flopy disk mechanism. A 720k (formatted) 3.5" mechanism is always fitted as drive A: and some units also have a second fitted as drive B: . All units have an external disk drive connector allowing the fitting of any one of the following units: 720K 3.5", 1.4MB 3.5", 360K 5.25". There is a switch which enables the user to assign the external drive as Drive B: . On a one-drive system Drive B: would otherwise share the internal Drive A: and on a two drive system Drive B: would otherwise be the right-hand drive.

If you are using an external disk drive then your PC should be told, as there may be undesirable side-effects if you rely on MS-DOS's automatic diskette recognition abilities (only implemented in version 3.2 and above). When the correct parameters are set then the FORMAT and DISKCOPY commands will operate correctly, the former writing the correct number of tracks and the latter automatically checking that a particular drive combination is legal.

In addition, it is possible to 'swap' the designations of Drive A: and Drive B: so that the external drive appears (to most software) to be Drive A:. In particular, this will allow the booting of software from the external drive.

Changeline

This is a facility which allows your PC can tell if the diskette has been changed by interrogating special circuitry on the drive mechanism. It results in a performance increase. Normally a 3.5" drive has the changeline feature, but a 5.25" does not. If in doubt, assume the drive does not.

It is particularly important that the changeline parameter is set correctly in the PC's NVR. The built-in 3.5" drive has a changeline facility and it will be assumed that Drive B: also has that facility unless informed otherwise. If the drive does not have changeline, when your PC thinks it has, then the drive will be unusable. MS-DOS 3.2 will not allow a 3.5" drive without changeline.

Setting up the External drive

The preferred method is to alter the values stored in your PC's NVR (Non-Volatile RAM).

Additional drive type	Command
360K 5.25" no changeline	**DEVICE SETFD /d:1 /n:0 -c**
720K 3.5" with changeline	**DEVICE SETFD /d:1 /n:1 +c**
1.4M 3.5" with changeline	**DEVICE SETFD /d:1 /n:2 +c**

Adding the parameter **+s** to any of the DEVICE SETFD commands will cause swapping of drives A: and B: . The designation **/d:1** is retained, however.

IMPORTANT:
FOR ANY OF THE ABOVE CHANGES TO TAKE EFFECT, YOUR PC MUST BE RE-BOOTED.

To re-boot, either press **[Ctrl][Alt][Del]** or add the **/b** parameter to the last DEVICE command.

DRIVPARM

This method of informing MS-DOS of changes to the form-factors of floppy disk drives is wholly redundant due to the more sophisticated facilities provided by the NVR parameters. If you do use DRIVPARM, then the commands below must be used. Particular care should be given to the consistency of the **/c** changeline parameter. The drive number used is 0 for A: and 1 for B: - calculated after the effect of any DEVICE **+s** command. The DRIVPARM command will over-ride any form-factor information held in NVR.

The new command line must be added to the MS-DOS CONFIG.SYS file; using an editor such as RPED.

Additional drive type	Command
360K 5.25" no changeline	**DRIVPARM = /d:1 /f:0**
720K 3.5" with changeline	**DRIVPARM = /d:1 /f:2 /c**
1.4M 3.5" with changeline	**DRIVPARM = /d:1 /f:7 /c**

IMPORTANT:
FOR ANY OF THE ABOVE CHANGES TO TAKE EFFECT YOUR PC MUST BE RE-BOOTED.

To re-boot, either press **[Ctrl][Alt][Del]** or add the **/b** parameter to the last DEVICE command.

Programming the Floppy Disk

The disk will normally be accessed by opening files using the Disk Operating System. Consult the reference publications for that software. If it is required to read and write directly to the disk then the following functionality is provided:

Interrupt 19: Disk I/O.

This software interrupt provides disk read, write, verify, and format functions for the drives fitted to the standard floppy disk controller.

For all disk functions the Carry Flag (CY) will be clear if no error else it is set if an error (and AH = error number). All other flags are corrupt.

For all disk functions except 0 & 1 drive number (DL) is checked and if greater than maximum drive number the function is rejected and return status is set to 1 (AH=1) and carry is set.

00h	Operation completed successfully.
01h	Incorrect Function (or drive) specifier.
02h	Missing address mark error.
03h	Disk write protected (Write or Format commands only).
04h	Record not found.
06h	Media Changed.
08h	DMA overrun error.
09h	Attempted DMA over a 64K segment boundary.
0Ch	Non-supported Drive type.
10h	CRC error.
20h	Floppy disk controller error.
40h	Seek error.
80h	Floppy disk controller timeout (Drive Not Ready).

Disk Int 19 function 0: Initialize Disk Sub-System.

This Function performs a total initialization of the disk interface as follows:

1. Reset the FDC (Floppy Disk Controller).

2. Re-configure the FDC parameters to those specified in the disk parameter table (interrupt vector 30 at 0:78h - 7Bh)

Entry: AH = 0

Exit: AH/Flags = Status as specified above.
All registers preserved.

When an error is returned by any other Diskette I/O Function, the Initialize Disk Function should be called prior to the next disk I/O operation.

Disk Int 19 function 1: Return Last Status.

This function returns the status byte and Carry Bit of the last disk I/O operation.

Entry: AH = 1

Exit: AH/Flags = Status of last disk I/O as specified above.
All other registers preserved.

Disk Int 19 function 2: Read Sector.

Entry: AH = 2
DH = Head Number (0 or 1)
DL = Drive Number (0 or 1)
CH = Track Number
CL = Starting Sector Number
BX = Offset Address of Read Data Buffer
ES = Segment Address of Read Data Buffer
AL = Number of Sectors to Read (All on same track)

Exit: AH/Flags = Status as specified above
AL = Number of Sectors successfully read
(Corrupt if Timeout error)
All other registers preserved

Disk Int 19 function 3: Write Sector.

Entry: AH = 3
DH = Head Number (0 or 1)
DL = Drive Number (0 or 1)
CH = Track Number
CL = Starting Sector Number
BX = Offset Address of Write Data Buffer
ES = Segment Address of Write Data Buffer
AL = Number of Sectors to Write (All on same track)

Exit: AH/Flags = Status as specified above
AL = Number of Sectors successfully written
(Corrupt if Timeout error)
All other registers preserved

Disk Int 19 function 4: Verify Sector.

Entry: AH = 4
DH = Head Number (0 or 1)
DL = Drive Number (0 or 1)
CH = Track Number
CL = Starting Sector Number
AL = Number of Sectors to Verify (all on same track)

Exit: AH/Flags = Status as specified above
AL = Number of Sectors successfully verified
(Corrupt if Timeout error)
All other registers preserved

Since the verification process is halted upon the first occurrence of an error, AL represents the number of sectors successfully verified prior to the occurrence of an error or total sectors verified if no error.

Disk Int 19 function 5: Format Track

Entry: AH = 5
DH = Head Number (0 or 1)
DL = Drive Number (0 or 1)
CH = Track Number
BX = Offset Address of Format Buffer
ES = Segment Address of Format Buffer

Exit: AH/Flags = Status as specified above.
All other registers preserved.

The format buffer contains four bytes of information for each sector on the track:

1. Track Number
2. Side Number
3. Sector Number
4. Sector Size Code:
   - 0 - 128 Bytes/Sector
   - 1 - 256 Bytes/Sector
   - 2 - 512 Bytes/Sector
   - 3 - 1024 Bytes/Sector

The gap length, filler byte and sectors per track required by the FDC Format command are obtained from the disk parameter table.

Disk Int 19 function 8: Read Drive Parameters

Entry: AH = 8
DL = Drive Number (0 or 1)

Exit: AX = 0
BX = Drive Type. (1 - 360kb, 3 - 720kb or 4 - 1.44Mb)
DH = Number of Heads - 1. (1)
DL = Number of Drives installed. (1 or 2)
CH = Number of Tracks - 1. (39 or 79)
CL = Number of Sectors per Track. (9 or 18)
ES:DI = 20-bit pointer to Disk Parameter Table for the drive.

Flags corrupt and CY clear.
All other registers preserved.

Note that the drive type is determined by an I/O read of the FDC DataRate Register.

Disk Int 19 function 21: Read Drive Type.

Entry: AH = 15h
DL = Drive Number (0 or 1).

Exit: AH = 0 - Drive not present
1 - Diskette without change line present
2 - Diskette with change line present

Flags corrupt and CY clear.
All other registers preserved.

Disk Int 19 function 22: Read Disk Changeline.

Entry: AH = 16
DL = Drive Number (0 or 1).

Exit: AH = 0 - change line not active & CY clear
1 - invalid drive number
6 - change line active & CY set

Flags corrupt and CY as above
All other registers preserved.

Disk Int 19 function 23: Set Drive Type.

Entry: AH = 17h
DL = Drive Number (0 or 1).
AL = Drive Type 4 - 720 Kb drive
5 - 1.44Mb drive

Exit: AH = Status - CY set if error.

Flags corrupt and CY as above
All other registers preserved.

Disk Int 19 function 24: Set Media Type.

Entry: AH = 18h
DL = Drive Number (0 or 1).
CH = Number of Tracks - 1. (39 or 79)
CL = Number of Sectors per Track. (9 or 18)

Exit: AH = Status - CY set if error.
ES:DI = 20-bit pointer to Disk Parameter Table for the media type.

Flags corrupt and CY as above
All other registers preserved.

DISK PARAMETER TABLES

The following Disk Parameter Tables are to be used for the two possible data rates supported by the floppy disk hardware.

Byte	720KB	1.44Mb	Function
0	DF	AF	FDC Specify byte 1 (Step Rate/Head Unload Delay)
1	02	02	FDC Specify byte 2 (Head Load Delay/FDC DMA mode)
2	64	64	Motor Off Timeout (5.4 sec)
3	02	02	Sector Size Selector (512 bytes)
4	09	12	Sectors per Track
5	2A	1B	R/W Gap Length
6	FF	FF	Data Length
7	50	6C	Format Gap Length
8	F6	F6	Format Filler byte
9	0F	0F	Head Settling Delay (15Ms)
A	04	04	Motor on Delay (500 Ms)

9. MAINTENANCE

9.1 Changing the batteries

The batteries powering the Battery-Backed RAM will need to be replaced from time to time. You have two options:

- either to wait for your PC to report that the batteries have gone flat before changing them
- or to change them on a regular basis, say once a year, before they go flat.

If you wait for your PC to tell you that they are flat, you will have to go through the procedure of setting the clock and setting the other details held in the battery-backed RAM each time you have to change the batteries.

The second option allows you to change the batteries without losing the information stored in the RAM, and this is the one we recommend: you should not need to change the batteries more often than about once a year, but it does depend on how much time your PC spends switched off.

The procedure in either case is as follows: it should be carried out either before you switch on your PC or after you have finished working with your PC and have switched off.

1. Start with the System Unit and Display mains plugs out of the supply sockets.
2. Lift up the Display so that its base comes out of the recess in the top of the system unit and then put it down to the righthand side of the System Unit.

Important: You must not move the Display in this way with the power connected to the System Unit or the Display.

It should be possible to move the Display without disconnecting the video cable which connects the Display to the System Unit; if necessary, unplug the cable before you move the Display and then reconnect once it has been moved.

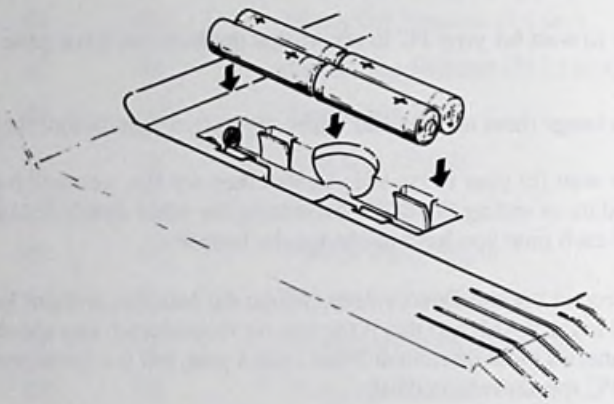
3. Plug both the System Unit and the Display into the mains supply. Switch the System Unit on by pressing the Power switch on the lefthand panel. Switch the Display on by pressing the Power switch on the back.

This provides power to the system which ensures that none of the information stored in the non-volatile RAM is lost while the batteries are changed.

4. Take the four batteries out of the battery compartment and throw them away.

5. Replace these batteries with four new AA batteries.

   Slot the four AA batteries into the battery compartment on the top of the System Unit as shown below. Check that you have each battery the right way round, with its positive (+) end over the plus sign inside the battery housing.



6. Hold down [Ctrl] and [Alt] and press [Del], and wait until the startup message appears.

7. Switch off by pressing the Power switches on the back of the Monitor and the lefthand panel of the System Unit. Unplug both the System Unit and Display from the mains supply.

8. Lift up the Display and place it back on the top of the System Unit, with its base fitted into the recess in the top of this unit.

Important: You must not move the Display in this way with the power connected to your System Unit or Display.

Again, it should be possible to move the Display without disconnecting the video cable.

The PC is now ready once again for normal use.

9.2 Cleaning

We recommend the use of non-CFC (chloro-flouro carbon) aerosol anti-static foam cleaners to clean both the screen and the plastic case. Under no circumstances should spirit-based cleaners be used.

PART TWO

GUIDE TO USING MS-DOS

STRUCTURE OF PART 2

This part of the manual aims to provide the information you need when you first start using MS-DOS on your PC, as well as being a source of reference when you have had experience of using MS-DOS.

Chapters 1, 2 and 3 introduce you to MS-DOS: explaining what it is, and how to use MS-DOS commands.

Chapters 4, 5, 6 and 7 describe what you can use MS-DOS for. Each chapter deals with a different aspect of using your computer:

- **Tailoring your PC to your needs**
- **Processing disks**
- **Organising your work**
- **Running programs**

The above four chapters describe how to set about achieving the result you require and tell you about the commands you will need to use. You should then look up the relevant command in Chapter 8, where all the MS-DOS commands are listed alphabetically.

When you first start to use MS-DOS, you will need to use these chapters to find out which commands you want to use. As you become more familiar with MS-DOS, you will be able to refer to the list of commands straight away, for detailed information on the command lines you have to type.

CONTENTS

Structure of Part 2

1. INTRODUCING MS DOS

Your PC will not be able to run programs, nor will it understand your instructions, without a disk operating system. MS-DOS is the disk operating system supplied with your PC.

Once installed on your PC, MS-DOS will translate the commands you type in, into instructions that your PC's processor understands. That is, you will use MS-DOS to communicate with, and control your PC.

You will be able to use these commands to instruct your computer to:

- process disks
- organise your information
- run programs
  simple programs; series of programs
- install and run application software
- handle the PC's input and output devices
- personalise the PC and make use of the PC's special features.

1.1 Starting to use MS-DOS

Before you can use MS-DOS on your PC, it has to be installed correctly.

If your PC has only floppy disk drives:

The MS-DOS System Disk is already installed for use with your PC and its keyboard.

Part 1 of this manual describes how to load MS-DOS into your PC. See Chapter 3, steps 6 and 7, "The Startup Procedure" and "Loading MS-DOS".

If your PC was supplied with a hard disk:

MS-DOS should already be installed on the hard disk.

Part 1 of this manual describes what you expect to see on the screen after you have switched on your PC.

If your PC has been upgraded to include a hard disk, or for some reason the supplied hard disk does not contain an installed MS-DOS, you should follow the instructions in Part 1, Chapter 3, "Installation".

1.2 Using MS-DOS commands

Internal and External commands

Some MS-DOS commands are already built into your PC and are contained in a file called COMMAND.COM. These are called INTERNAL commands. The rest of the commands are stored in files on your PC disks; these are called EXTERNAL commands.

Command lines

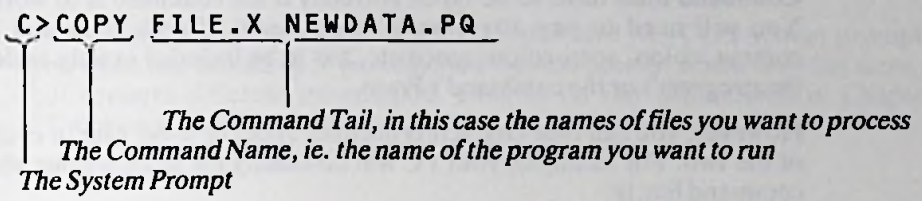
You use MS-DOS by giving it instructions called command lines. These command lines each contain:

- **the name of the program to be run** - the command name
- **the names of the files you want the program to process**
- **any other information the program needs**

} - the command tail

You use command lines to run programs you buy, such as word processors and spreadsheets; programming languages such as BASIC; text editors for editing your file; computer games; or advanced computing tools such as assemblers. You also use command lines to call up the MS-DOS commands that help you run other programs and organise data.

For example, to COPY a file called FILE.X to a file called NEWDATA.PQ, you would use the command line:



The **System Prompt** is put up on the screen by MS-DOS to show that it is ready to receive a command. The rest of the command line you type yourself. It automatically appears to the right of the system prompt.

What you type depends on the program you want to run or the command you want to use. The details are given by the program's or the command's **Form** or **Syntax**. The Form of each MS-DOS command is given in this manual as part of its detailed description. Command descriptions are listed alphabetically in Chapter 8. The Form of the command lines you need, to run any commercial programs you buy, will be given in the program's own user guides.

Within the command line, you give all the information MS-DOS needs to do the job you want. In particular, you must remember to include details of where the program that you want to run is stored, that is, in which drive and in which directory. You must also give the location of the files on which you want to work; the drive and directory where they are stored.

If you don't tell MS-DOS where to find these files it will assume that they are to be found in the default drive, among the files you are currently working on.

The location information you give is:

- the drive, if the file is not in the default drive
- the path to the directory if the file is not in the current directory of the drive (see Section 7.1)

If any of these terms seem strange, see Chapter 6.

Typing command lines

Command lines have to be typed correctly if the command is to work properly. You will need to pay attention to every detail of each command line: every comma, colon, semi-colon, space etc. has to be included exactly as described by the program's or the command's Form.

However, you can type characters in either upper or lower case or even a mixture of the two. For example, your PC will do exactly the same actions whether your command line is:

```
A>COPY FILE.X NEWDATA.PQ
A>copy file.x newdata.pq
A.COPY file.x newdata.PQ
```

Most mistakes will be picked up when MS-DOS starts to process the command line (see "When commands fail", later in the chapter), but there is always a chance that MS-DOS will do what you say, rather than what you want! If you spot a mistake while you are typing in the command line, use the [←] key to rub out what you have typed back to the mistake and then type the remainder of the command line in again.

When you have finished typing the command line, you press the [↵] key. This sends your instruction to MS-DOS for processing. The program you specified in the command line is then run. When the program has finished, MS-DOS puts a fresh system prompt on the screen. It is then ready to process another instruction for you.

More about the System Prompt

The main job of the system prompt is to tell you that MS-DOS is ready to receive a new command. If there is a system prompt on the screen with the cursor to the right of it, you can type in your command.

The system prompt also tells you which drive is the current default drive. MS-DOS can be set up to handle up to 26 drives at the same time - though it is normally set up to handle five. It calls these drives Drive A, Drive B, ... Drive E. The default drive is the one it assumes you want to use unless you tell it otherwise. When the system prompt is A>, the default drive is Drive A - your floppy disk drive (or the lefthand disk drive if you have two).

If you have two floppy disk drives, you might change the default drive to Drive B (the righthand disk drive); the system prompt would then become B>.

If you have a hard disk, and change the default drive from Drive A (the lefthand drive) to the hard disk drive, Drive C, the system prompt would then become C>. (How to change the default drive is described in Chapter 7 'Running programs'.)

The system prompt described here is the standard MS-DOS system prompt. If you wish, you can set up a 'personalised' system prompt which does the same job but contains different information. How to do this is described in Chapter 4 "Tailoring your PC to your needs".

More about the Command Name

The command name tells MS-DOS which command you want to use or which program you want to run. This could be:

- an MS-DOS internal command
- an MS-DOS external command (stored on the MS-DOS Disk)
- an MS-DOS or PC-DOS program
- a batch file (see Section 7.4 'Setting up a sequence of commands')

You will be able to recognise which of your files contain programs you can run from their filetype. MS-DOS programs and PC-DOS programs (often described together as DOS programs) have the filetype COM or EXE (ie names like MYPROG.COM or MYPROG.EXE) and Batch files have the filetype BAT (for example, MYBATCH.BAT).

The command name has two parts:

- first, the location of the file holding the program or the command
- second, the name of the file

The location part of the command name is only included when the command or program is not immediately available. MS-DOS internal commands are always available: the other programs that are immediately available are the MS-DOS external commands and other programs in the group of files you are currently working on, in your default drive.

For all other commands and programs, you have to tell MS-DOS in which drive and/or in which directory to look for the program. For example, if you want to use the external command DISKCOPY which you have stored on the built-in disk (either a hard disk, if your system has one of these, or an area of your computer's memory that can be used as a disk), you would start your command line:

```
C:DISKCOPY ...
```

If it were stored in a directory called COMMANDS on Drive C, you might start your command line:

```
C:\COMMANDS\DISKCOPY ...
```

The location part of the command name is often left out when the 'Form' or 'Syntax' of the command is given, which means you must remember to include this information if it is needed. Details of how to include the location in the command line are given in Section 7.1.

More about the Command Tail

The information a program needs in the command tail, and the order in which this is given, depend on the program. It is all laid out in the 'Form' or 'Syntax' statement for the command line and must be followed exactly if the program is to work correctly.

The Form of the command line for each MS-DOS command is given in this manual in Chapter 8, where all the commands are described in detail. If you buy commercial programs, the form of the command lines you need to run the programs will be given in the accompanying user guides.

The FORM statement indicates what file details you need to give to MS-DOS, to make the programs process the files. For example:

FORM `CHDIR` *[d:][\]path*

So, `CHDIR` is the MS-DOS command

*[d:][\]*path are the location details that the command needs.

It also tells you how to specify the program options you want to take advantage of. It does this by presenting items like *d: filename, parameter, and physical-device* in a different style to the rest of the line. (In this manual, these are always written in an italic script.) These are known as placeholders and they show where to insert details of the actual job you want to do, for example the name of the file you want to copy or the name of the file you want to store the copy in.

For example

`RENAME` *old-name new-name*

tells you to type **RENAME**, followed by a space, followed by the old (ie. the current) name of the file you want to rename, followed by another space, and finally the new name you want to give the file. So if you wanted to rename the file FILE.X and call it MYFILE.X1, the command line you would type would be:

```
RENAME FILE.X MYFILE.X1
```

The details, required to replace the placeholders, should be explained either in the notes describing the command line or in a general section covering the conventions used in the manual. There may also be parts of the command line that you can miss out because they are not appropriate. Details of the conventions used in this manual are given in Chapter 3 'Conventions'.

When Commands Fail

Commands fail because:

- you have made a typing error in the command line
- you haven't specified the location of a file correctly
- you haven't got enough memory space for the program to run
- you haven't got enough room on the disk to store the new files the program produces
- the file holding the program has been corrupted (or it never worked in the first place!)

You can usually tell why a command has failed from what appears on the screen.

If MS-DOS puts up the message "Bad command or file name", it hasn't found the command, program or batch file you wanted to run.

The commonest reason for this is that you mistyped the command name. For example, you might have typed COYP instead of COPY. The other possibility is that the program file is not in the directory or directories MS-DOS searched. You may have misdirected MS-DOS, for example because you forgot which disk you had in the drive. Check where you told MS-DOS to look for the file.

If MS-DOS puts up a message like "File not found" together with details of the file and a fresh system prompt, it hasn't found one of the files you asked it to process.

Either you mistyped the filename or the filetype or the file isn't in the directory you specified. Check where you told MS-DOS to look for the file.

In both of these cases, to try again, you can either retype the command line at the new system prompt or copy and then edit your previous command line. The keystrokes to help you do this are described in Section 7.2. When you have finished preparing your new command line, press the [↵] key.

If MS-DOS puts up some other message like 'Memory insufficient to run program' or 'Disk full', it has met some other problem in carrying out your command.

Turn to chapter 10 'Troubleshooting' where error messages like this are explained and then take whatever action is appropriate. You may, for example, need to erase some files you no longer need from the disk or change what you are trying to do.

If in doubt, consult your dealer.

If the screen goes blank or your PC generally seems 'dead', your program file may have been corrupted.

Reset your PC (see Part I, Chapter 3) or, if you can't reset your machine, release the disk(s) in the drive(s), switch off and then work through the Startup procedure again (see Part I). Then try running your program again. If you get exactly the same failure this time, try running your back up copy of the program (if you have made one of these). If that fails too, consult your dealer.

2. OVERVIEW OF MS-DOS COMMANDS

This chapter should give you a feeling for the sort of operations for which you can use MS-DOS commands, and get you used to the idea of typing command lines.

The chapter is split into two parts:

- A summary of Chapters 4-7, which describe the basic uses of MS-DOS
- A guided tour through some simple uses of MS-DOS commands

2.1 The tasks you can use MS-DOS for

Tailoring the PC to your needs

When you bought your PC, it was set up to be used in a very conventional way -with the keyboard as your main means of giving instructions and information (ie. input) to the computer, and the display screen used by the computer to display (ie. output) information. Among other details set up conventionally, are the system prompt and the number of lines and characters in each page of information.
each page of information.

This chapter explains how to tailor your PC to your needs by:

Setting up Input and Output Devices

Bring a printer or a communications link into play or change the details of how your current set of input and output devices work by:

- setting device parameters
- organising what information is sent where

Personalising your PC

- by setting up a personalised system prompt
- by giving your disks names by which you can identify them

Setting your PC's internal clock

- by setting the date
- by setting the time

You don't have to set the clock but if you do, the date and time MS-DOS automatically records with each file for you do actually tell you when the file was last changed. If you don't set the clock, the date and time recorded with each file can be very misleading.

Processing your disks

Processing your disks covers:

Preparing new blank floppy disks

- Dividing up the blank disk into sections so that your PC can store and retrieve information from the disk. This process is called Formatting the disk.

Maintaining existing floppy disks

- Making duplicate copies of your disks so that you have a reserve or 'back up' copy for use in case of accidents
- Checking that the data stored on a disk has not become corrupted by 'Verifying' the contents of the disk.

These tasks are fundamental to using floppy disks in your PC.

Organising your work

This chapter describes how to make and keep your files readily available for use by:

Putting your files into groups

As you increase the number of files on a disk, you will find that they become much more manageable and more convenient to use if related files are grouped together. MS-DOS then lets you work on just a group at a time.

The first section of this chapter describes how to

- Start new groups
- Get rid of groups you no longer need

Disk housekeeping

The selection of files you have on your disks and the way these files are grouped will frequently become out of date. You may well have files you no longer need taking up valuable space on your disk.

The commands in this section help you put this right by

- Creating new files
- Making copies of existing files
- Erasing files you no longer need
- Changing what files are called

Protecting important files

Files are readily erased or overwritten. This section describes how to protect your most valuable files - programs you have bought, last year's accounts, etc - against being erased or overwritten accidentally.

Running programs

This chapter describes the primary job of your computer - to run programs for you. In particular, it describes ways to run programs more efficiently through:

Shortcuts at the keyboard

- Editing the previous command line to save typing
- Stopping the program that is being run. Sometimes you can then make it continue where it left off.
- Keeping a record on your printer of what you have typed and the programs you have run
- Extending the number of disks and directories MS-DOS will search automatically to find the program or data file you want

Redirecting input and output

Programs usually take their input from the keyboard and send their output to the display screen. This section shows you how to

- tell the program to take its input from a file or from a telephone link
- tell the program to send its output to a file, or to a printer or down a modem link
- link programs together so that the output of one program is the input of the next program in the chain

Setting up a sequence of programs

- to run one after another automatically
- to be run whenever your PC is turned on

2.2 Some practice in using MS-DOS commands

The rest of this introduction takes you through some simple uses of MS-DOS. Although you can just read it, we recommend you work through the instructions on your PC as it will get you used to using MS-DOS commands.

The sets of commands we will use are those for finding out about the files stored on your disks and those for keeping your files organised. These are described in Sections 7.1 and 6.2. We recommend that you look up the commands used in the reference sections as you work through this introduction. This will help you get used to the way commands are described in the reference chapters and so make these easier to use when you start using MS-DOS commands in earnest.

Note In this section, we remind you to press the Carriage Return key [↵] to send each command you type to MS-DOS by putting [↵] at the end of each command line. The more advanced sections of this manual assume that you know to press the [↵] key at the end of each command line.

Remember to type command lines only when the last thing displayed on the screen was a system prompt (eg >) with the cursor to its right.

Simple Commands

MS-DOS commands are set up so that unless you tell them otherwise, they will work with the files in the default directory, that is, the directory you are currently working on. On Startup, or immediately after you have reset your PC, the default directory is the main or 'Root' directory on Drive C (or Drive A if you do not have a hard disk).

Often, the first thing you will want to do is to find out what files are held in this directory. (Section 7.1 describes how to do this). The command to use to display a directory is the DIR command and to display all the files in the default directory, you just type:

```
DIR [↵]
```

You can type DIR in either upper or lower case letters, or a mixture of the two: it makes no difference to the result. Something like this will appear on your screen:

```
 Volume in drive C is PC2086UK1
 Directory of  C:\

COMMAND  COM     25276 24-07-87  12:00a
ANSI     SYS      1647 24-07-87  12:00a
CONFIG   SYS        18 27-07-88   2:58p
COUNTRY  SYS     11254 24-07-87  12:00a
DRIVER   SYS      1165 24-07-87  12:00a
KEYBOARD SYS     19735 24-07-87  12:00a
PRINTER  SYS     13559 24-07-87  12:00a
RAMDRIVE SYS      6481 24-07-87  12:00a
AUTOEXEC BAT        58 27-07-88   2:58p
ASSIGN   COM      1530 24-07-87  12:00a
BACKUP   COM     29976 24-07-87  12:00a
CHKDSK   COM      9819 24-07-87  12:00a
COMP     COM      4183 24-07-87  12:00a
DEBUG    COM     15866 24-07-87  12:00a
DISKCOMP COM      5848 24-07-87  12:00a
DISKCOPY COM      6264 24-07-87  12:00a
EDLIN    COM      7495 24-07-87  12:00a
FDISK    COM     40919 24-07-87  12:00a
FORMAT   COM     11671 24-07-87  12:00a
GRAFTABL COM      6136 24-07-87  12:00a
GRAPHICS COM     13943 24-07-87  12:00a
```

This gives you the names of all the files in the default directory.

It can also be used to give information about particular files by adding a 'command tail' to our original file. For example,

```
DIR COMMAND.COM [↵]
```

gives you the standard information, but just about the file COMMAND.COM. You might use a command like this to check whether a paricular file is in the default directory. If the file is not found, MS-DOS will display the message:

```
File not found
```

DIR is one of a small group of commands which will process a number of files at a time, if you wish. You tell the command to do this by putting a special sort of file name into the command tail - one that includes 'Wildcard' characters. These wildcard characters represent any valid character or group of characters. The special file name is used as a 'template' with which the name of each of the files in the directory is compared. Only if the name matches the template will information on the file be displayed.

The special 'Wildcard' characters are * and ?. An asterisk in the template represents any valid group of characters that can appear in that position in the template. A question mark only represents a single character.

So, for example, if you wanted to list all the files with the filetype COM, you would use the template ***.COM** and the command:

```
DIR *.COM [↵]
```

And if you wanted to list all the files, you might type:

```
DIR *.*  [↵]
```

In fact you would just type **DIR** because **DIR** is a recognised shorthand for **DIR *.***. Similarly, you probably wouldn't type **DIR *.COM** to tell MS-DOS to list all files with the filetype COM because you can just type **DIR .COM**.

If you wanted to list all the files with the filetype COM whose filenames started DISK, you might use the template **D*.COM** and the command:

```
DIR D*.COM [↵]
```

This will, however, give you all the files that match this template, ie every file that has a filename starting with the letter D - not just the ones with the filenames starting DISK - as you will see if you try this out.

Longer commands

Some commands need the name of more than one file in the command tail. For example, when you want to make a copy of one of your existing files, you need to tell MS-DOS both the name of the file you want to copy (known as the 'Source') and the name you are going to give the copy you create (known as the 'Destination'). For example, the command line to make a copy of the file AUTOEXEC.BAT (the Source) and store it as the file MYBAT.BAT (the Destination) is:

```
COPY AUTOEXEC.BAT MYBAT.BAT [↵]
```

The Source file is not changed in any way by being copied.

As you don't need the file MYBAT.BAT, you might as well delete it with the command:

```
DEL MYBAT.BAT [↵]
```

Commands with options

The commands we have typed so far have all been as simple as possible. You can get commands to do more for you by using command options. These options are listed in the command specifications in slanting brackets to show that you don't have to include these every time you use the command.

The options are often represented by a slash followed by a letter. For example, the DIR command has a /W option that prints the names of the files in a directory five to a line, rather than in a long list. This way of displaying the directory uses command lines like:

```
DIR /W   [↵]          - the whole directory
DIR .EXE /W [↵]       - all the files with the filetype EXE
```

Another possibility with the DIR command is to use the /P option to make MS-DOS pause at the end of each screenful so that information isn't scrolled off the top of the screen before you have a chance to read it. You can use this option on its own or you can use both options at once, by typing:

```
DIR /W/P [↵]
DIR /P/W [↵]
```

the order in which you list the extra options usually doesn't matter: if it does matter, this will be emphasised in the description of the command.

Another drive, another directory

So far in this tour, we haven't looked outside the default directory. The final commands we will look at in this section show what you need to add in when you want to process a file that isn't in the default directory.

A very simple example of this is using the DIR command to see if a particular file is in the current directory on Drive A (the floppy disk) without first changing the default drive to Drive A. If you type:

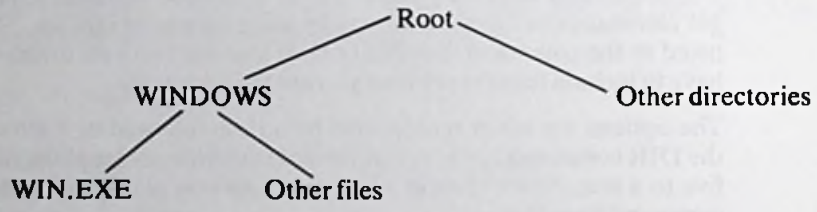
```
DIR A:COMMAND.COM [↵]
```

the PC will display similar details to those displayed if you typed:

```
DIR COMMAND.COM [↵]
```

the only difference being that instead of giving details of a file in the current directory on the default drive, the PC gives details of a file called COMMAND.COM in the current directory on Drive A (if such a file exists).

To use files which aren't in the current directory, you have to include details of how to get to this directory. For example, suppose the file you want details of is called WIN.EXE and it is stored in a directory called WINDOWS that branches off the Root directory on Drive C



The path to the file is therefore: Root → Windows → WIN.EXE which you write as **\WINDOWS\WIN.EXE** (the first backslash telling MS-DOS to start the path from the Root directory). Drive C is the default drive, so your DIR command line becomes:

```
DIR\WINDOWS\WIN.EXE [↵]
```

If drive C hadn't been the default drive, this command line would have been:

```
DIR C:\WINDOWS\WIN.EXE [↵]
```

Postscript

This brief tour should have given you an idea of how to use MS-DOS commands to carry out tasks for you. Chapters 4-7 give detailed descriptions of the tasks and the commands you will need to use.

3. CONVENTIONS

The following tutorial chapters make great use of placeholders to describe the form of the various command lines. This means that where you should put the actual name of, say, a file or a drive, we have put a brief description of the item printed in italic (ie. slanted) text - with hyphens between the words if more than one word is used to describe a single item.

For example, if you want the command to process a file called MYFILE.TXT, you should put `MYFILE` where we have put the *file name* and `TXT` where we have put *filetype*.

Most of the placeholders are explained alongside the command itself but the most common ones are explained in Section 3.1.

In addition, optional parts of the command line are surrounded by slanting square brackets: these parts may be left out altogether.

For example:

`DATE` *[dd-mm-yy]*

means that depending on the precise details of what you want to do, you either type:

`DATE` *dd-mm-yy*

or just

`DATE`

A special example of this is *[\]* where this precedes path. If your chosen path starts at the root directory on the drive, you should include the backslash but if it starts at the current directory, you should leave the backslash out. (For further details, see Section 7.2.)

Anything not printed in italic must be typed in exactly as shown.

If some section of the command line can be included once or included many times, the placeholder will be given twice, followed the second time by ...

For example:

PATH *\path[;\path...]*

is used to represent any of

PATH *\path*
PATH *\path;\path*
PATH *\path;\path;\path*

etc.

Another convention is that where you have to choose just one out of a number of options, the options are written out in full but they are separated by solid vertical lines. For example:

ON|OFF

means type in either **ON** or **OFF** but not both. Do not type the **|**.

Take care to separate the items exactly as specified in the description of the command line. If this gives a single space or a comma between two items within the command line, then you must put one or more spaces or a comma between these two items.

Whenever you are not sure how to interpret the Form statement for a particular command, study the examples given alongside the command. These should clear up any uncertainty for you.

A further convention used in this manual is to represent 'Control codes' (that is, the special codes that control, for example, clearing the screen) as Ctrl-*character*. Some of the function keys on the keyboard may be set up to produce these codes but you can always produce them by holding down the **[Ctrl]** key and pressing the appropriate character key. When a control code is displayed on the screen, it will normally be shown as ^ *character*.

3.1 Standard placeholders

command	a complete command line, here incorporated in another command line
d	the drive holding the file you want to use represented as a letter. Often followed by a colon: for example, *d*: can be replaced by A:, B:, C: or ...
destination	the name of the file you want the copy to be stored in, written as *filename.filetype* (see below), or the name of a logical device you want the contents of a file to be sent to. (See Section 6.2)
device	the name MS-DOS recognises for an input or output device attached to your PC. (See Section 4.1).
filename	the first part of a file's name. For example, if the file's name is MYFILE.TXT, then the *filename* is MYFILE
filetype	the second part of a file's name. For example if the file's name is MYFILE.TXT, then the *filetype* is TXT
n	a number. Details of the range of numbers that can be used will be given in the description of the command
path	the full details of the path from either the current directory or the root directory on the drive to the directory you want to work with, starting with the name of the first directory away from your starting point and finishing with the name of the directory you want to work with. The different directory names should be separated by single backslashes. For example, the path between the root directory and a directory called SUBDIR1 might be DIR1\SUBDIR1 (See Section 7.2)
source	the name of a file you want to copy written as *filename.filetype* (see above) or of a logical device from which you want to take information and store it in a file. (See Section 6.2)
source-drive	the drive holding the disk you want to copy, written as *d*: (see above).
string	a string of characters of a parameter that has been set to a string of characters
target-drive	the drive holding the disk you want to store the copy on, written as *d*: (see above).

4. Tailoring your PC to your needs

When you bought your PC, it was set up to be used in a very conventional way -with the keyboard as your main means of giving instructions and information (ie input) to the computer and the display screen used by the computer to display (ie output) information. The system prompt and the number of lines and characters in each page of information are among other details set up conventionally.

This chapter describes how to set up your PC to work in precisely the way you want.

You might, for example, want to set up your PC so that it can send the data it normally sends to the printer down a communications link to another computer. You might want to change the number of lines and columns on your screen so that you can run a program that was designed to run on, for example, a 40-column screen.

Other possibilities are to add extra information to your system prompt such as the current directory or the time and giving your disks names that help you remember what information they hold.

This chapter explains how to tailor your PC to your needs by:

- **Setting up Input and Output Devices**

  Bring a printer or a communications link into play or change the details of how your current set of Input and Output Devices work by:

  - setting device parameters
  - organising what information is sent where

- **Personalising your PC**
  - by setting up a personalised system prompt
  - by giving disks labels

- **Setting your PC's clock**
  - setting the date
  - setting the time

4.1 Setting up your PC's Input and Output Devices

Your PC is ready, set up:

- **to receive most instructions and information from the keyboard**
- **to send program output to the monitor (unless told otherwise)**
- **to send printer output to the Parallel Printer port on the back of your machine**
- **to send and receive any auxiliary information via the Serial Interface on the back of your PC**

This is a perfectly satisfactory arrangement, but there are two possible ways in which you might want to change it:

1. **You might want to change the roles of some of the Input and Output Devices.**

   You might, for example, want to make the Serial Interface the main information channel because you have linked your PC to another computer through this connector. Or you might need to send printer output to the Serial Interface because your printer has a serial interface to it, rather than a parallel one.

2. **You might want to replace an Input or an Output Device or you might want to change how the current device is set up.**

   For example, you might want to replace your printer or your communications link with a new one that needs data sent to it at a different speed or checks the data has been sent correctly in a different way - or you might just want to change the speed at which data is sent to your current link.

   The commands that set up the special needs of your system should be put into the AUTOEXEC.BAT file on the disk you use as your Startup disk, because you will need these commands every time you use your PC.

Changing the roles of particular devices

Your programs appear to be designed to take some information through the keyboard and other information from, say, the Serial Interface and to send some messages to the screen and some, say, to the printer. Changing where information comes from and goes to would seem impossible without rewriting the program.

In fact such changes are made very easily because, in general, your programs don't make use of your Input and Output Devices directly. Instead, they use 'logical' devices which MS-DOS links to 'physical' devices such as your keyboard and your monitor. If your program does use the Input and Output Devices directly, it would indeed have to be rewritten before information could be taken from or sent to other places.

Section 7.3 covers how to change where the 'keyboard' input comes from and where the 'monitor' output goes to, command by command. This section describes how to make those changes more permanent.

To change which device acts as the keyboard

>CTTY> Normally, all command lines are typed in at your PC's keyboard. However, if you link your PC to another computer or if you attach a terminal to your computer, you might well want to give your PC instructions from the other computer or from the terminal.

You probably will have connected the computer or the terminal to your PC via the Serial Interface on the back of your PC. So you need to tell MS-DOS to take instructions from the Serial Interface, not from the keyboard. This is done with a CTTY command. All this command needs to know is the MS-DOS device name for the Serial interface. MS-DOS has two different device names for the Serial Interface - COM1 and AUX, either of which can be used in the CTTY command.

The command has an immediate effect. Once you have entered your CTTY command and it has been accepted, the next command line must come from the new input device.

When you want to return to using the keyboard again, you simply use another CTTY command - this time giving it the device name CON, which is the name MS-DOS has for both the keyboard and the monitor. However, this command will have to be entered at your current 'keyboard'. If you get into difficulties, you always have a way out in either resetting your PC by holding down **[Ctrl]** and **[Alt]** and pressing **[Del]** (using your PC's keyboard) or switching off and switching on again. Either of these actions will, however, cause the loss of any unstored data in your PC.

Sending printer output to the Serial Interface

>MODE> Printer output from your programs is normally sent to the Parallel Printer port on the back of your PC. The chances are that you have a printer with a parallel interface to it.

However, if your printer has a serial interface to it, you will need to attach this to a serial port, for example the Serial interface on the back of your machine. You will also want MS-DOS to direct printer output to this port if you want to send this ouput to another computer.

The command to use is MODE. One of this command's jobs is to tell MS-DOS to send all information directed to a particular printer device (known to MS-DOS as LPT1, LPT2 or LPT3), to a serial device (known to MS-DOS as COM1 or COM2). The command takes effect immediately. LPT1 is set by default to the device attached to the Parallel Printer connector on the back of your PC. COM1 is set by default to the device attached to the Serial Interface on the back of your machine. (In this instance you can't use the device names PRN and AUX, often used for devices attached to these two connectors).

When you want to send this output to a parallel printer again, just use another MODE command - this time leaving out the serial device name but including any device settings your parallel printer requires, even if you had set these up before. Otherwise the default settings will be used.

New equipment or settings

>MODE> Your PC must be set up to communicate with each of its input and output devices. MS-DOS initially holds the details your PC needs to communicate with the devices with which it is supplied, but these details have to be changed when:

- **you install a new peripheral**

  For example, you might install a new printer or a new communications link.

- **you change the setting on an existing peripheral**

  For example, you might change the speed at which data is transmitted along a communications link.

- **you run a program that has been set up for a different screen**

Note: Remember if you want to use your printer to print graphics screen displays, use the GRAPHICS command (which is restricted to low resolution screen modes).

To set up a parallel printer

There are two aspects of a parallel printer that you may have to set: the maximum number of characters that can be printed on a line and the spacing between the lines. In addition, because your PC will send data to the printer faster than the printer can handle it, you may also want to tell MS-DOS to keep trying to send output to the printer.

MS-DOS has built into it some default settings which may well suit your printer. These are 80 characters per line and 6 lines per inch. The command to use to change either of these settings and to tell MS-DOS to keep trying to send output is the MODE command.

Just one MODE command is used to set all of these values. Any aspect you don't specifically set in the command will be set to the default value.

To set up a communications link

The aspects of a communications link you may have to set are:

- **the rate at which data is transmitted on the link (the baud rate)**
- **the checking that is applied to the data (the data's parity)**
- **the number of data bits to each character**
- **the number of stop bits to each character**

You may also need to set these parameters if you attach a serial printer, rather than a communications link to your Serial Interface - along with a special setting that tells MS-DOS that you are using the link in this way.

The values you need to set should be specified in the literature supplied with the communications link. If you can't find this information in the brochures, consult your dealer. (Full details of communications links and what these parameters describe are outside the scope of this manual).

MS-DOS has built into it some default settings for the Serial Interface on the back of your PC which may well suit your communications link. These are Even parity and 7 data bits. In addition, setting 110 baud automatically sets 2 stop bits but any other transmission rate sets 1 stop bit.

The command to use to change any of these settings and to tell MS-DOS that you are using the interface for a printer is the MODE command. Just one MODE command is used to set all of these values. You have to set the baud rate but any other aspect you don't specifically set in the command will be set to the default value.

Adjusting the screen display

Your PC normally has an 80-column display and either a colour or a monochrome screen. However, the programs you buy will have been set up for:

- **either a colour or a black and white display**
- **either a 40-column or an 80-column screen**

These programs will run perfectly well, but if they have not been set up for an 80-column colour screen, they may not give quite the screen output you expected. To correct this, you will need to set your PC's screen to match that expected by your program. The command to use for this is again the MODE command.

Unlike the case with some other PCs, changing the screen mode in this way won't affect the screen alignment and so no adjustment will need to be made.

4.2 Personalising your PC

Personalising your PC is about changing some aspects of the way your computer works so that you can set it up to suit your requirements.

Personalising the system prompt

>PROMPT> Your personal MS-DOS system prompt doesn't have to be the current default drive. It can contain any of the following:

- **the default drive**
- **the default directory**
- **the date**
- **the time**
- **the version number of your MS-DOS**
- **special text or characters**

- whatever information you will find most helpful while you are using MS-DOS.

The command that sets up a personalised prompt is the PROMPT command.

Your new style of prompt will be used until your PC is reset. Then MS-DOS will revert to the standard C> prompt. If you want to always use a special style of prompt, simply put the PROMPT command to set it up into your AUTOEXEC.BAT file (see Section 7.4).

Giving disks labels

>LABEL> When you have a number of files and directories on a disk, it is not always easy to tell which disk you are working with, particularly if it could be any one of a number that you haven't used for some time.

The way to make finding out which disk you have in the drive much simpler is to give each of your disks a disk label - that is, an individual name of up to 11 characters. This name is always displayed when you use the DIR command to list any directory on the disk, even ones way down the directory tree - giving you a very easy way to tell which disk it is. The disk label also gives you a useful title by which you can catalogue your disks and the files they hold.

Use the LABEL command (Section 8) to give a disk such a label or to change the label if it already has one.

4.3 Setting your PC's clock

>DATE>
>TIME> If your PC is fitted with an internal clock then it is maintained in the battery-backed area of the computer's RAM so that it keeps perfect time even when your PC is switched off. However, it will be reset to 1 January 1980 if the batteries start to go flat. You can avoid this ever happening by changing the batteries regularly - say, once a year - but if the batteries do go flat, you should first renew the batteries and then reset the clock.

You can reset the clock through the MS-DOS commands, DATE and TIME, described in Section 8, and in Chapter 3, Step 8, and in the Installation chapter (3) in Part one of this manual.

5. PROCESSING FLOPPY DISKS

Processing your floppy disks covers:

- **Preparing new blank disks**
  - Dividing up the blank disk into sections so that your PC can store and retrieve information from the disk. This process is called Formatting the disk.

- **Maintaining existing disks**
  - Making duplicate copies of your disks so that you have a reserve or 'back up' copy for use in case of accidents.
  - Testing your disks when you suspect that the reason for a program failing was a damaged disk.
  - Comparing a copy you have made of a disk against the original.

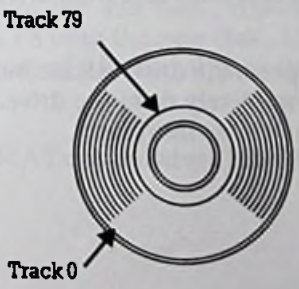
These tasks are fundamental to using floppy disks in your PC. For example, the storage area of each new blank disk you buy must be formatted before you can use it, even if you just want to store one program or data file on it. The process of copying a disk, automatically includes formatting the disk used for the copy. Formatting is also used to reclaim a disk on which the data has become electronically damaged.

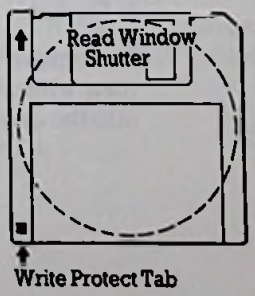
Formatting and copying disks are such important tasks that the steps you need to take have been included in this chapter.

5.1 Preparing new disks for use (Formatting)

>FORMAT > The process of marking out a new blank floppy disk uses the FORMAT command.

The simplest version of the command will give the new disk the number of tracks and sectors appropriate to the drive in which you format the disk.





To prepare a disk for use simply to store programs and data, follow the instructions below:

1 **Display a C> system prompt (or an A> if you have a floppy disk PC)**

Return to Drive C as the default drive. If you are not certain how to do this, reset your machine - instructions for this are given in Part 1, Chapter 3. If you have yet to switch on your PC, go through the Startup procedure described in Part 1.

2. **If you have a floppy disk PC place your MS-DOS disk in Drive A (your lefthand drive if you have two).**

This disk will already be in the drive if you have just switched on or reset the machine.

3. **Make the root directory the current directory.**

If you have just switched on or reset your PC, you need not do anything. If you have been using MS-DOS command lines, you may need to type:

```
CD\   [ ↵ ]
```

To return to working on the root directory.

4. **Type:**

```
FORMAT A: [ ↵ ]
```

After a short while, you should see the following message:

```
Insert a new diskette for drive A:
and strike ENTER when ready
```

5. **If you have a hard disk PC, insert the disk you want to format. If you have a floppy disk PC, remove the MS-DOS disk from drive A: and then insert the disk you want to format.**

To remove a disk, press the drive release button. This releases the disk, you can then withdraw it completely from the drive. Insert the disk you want to format into the drive until it latches into place.

6. **Press the [↵] key**

   The message: `Head: n  Cylinder:  nn` will appear on the screen.

   After a short while, messages similar to the following should appear on the screen:

```
Format complete
nnnn bytes of disk space
nnnn bytes available on disk

Format another (Y/N)
```

7. **Type [Y] [↵] if you want to format another disk; type [N] [↵] to leave FORMAT.**

   The FORMAT command can also be used with an old disk on which the data has been corrupted. If the disk hasn't been physically damaged, formatting the disk can reclaim all its storage space. However, before you do this, be sure to copy as many as possible of your files to other disks before you reformat the disk. Formatting a disk wipes it clean of any stored data.

   The options associated with the FORMAT command let you:

- **both format and prepare the disk for use as a Startup disk, i.e. as a disk that you can use to load MS-DOS after you switch on your PC**
- **format the disk so that it could later be made a Startup disk without losing any of the data and programs then stored on the disk**
- **give the disk a disk label**
- **have some non-standard number of tracks and/or number of sectors**

   When you specify that you want the disk to be used either now or later as a Startup disk - i.e. as a disk that you can use immediately after switching on your PC, FORMAT allocates some of the disk space to the system files needed to run MS-DOS and, to prepare the disk fully, copies the files COMMAND.COM, IO.SYS and MSDOS.SYS onto the new disk. The two .SYS files never appear when you list the directory of a Startup disk so that you don't accidentally delete them.

   Full details of the FORMAT command are listed in chapter 8.

>SYS> If you decide to postpone making your disk a Startup disk, you can copy IO.SYS and MSDOS.SYS to the disk later by using the SYS command.

The space allocated to the system files on a Startup or a Potential Startup disk cannot be used to store data files and program files, and so the amount of space available for your files is reduced. It would therefore be wasteful to make all your disks potential Startup disks. Instead, only choose this option if you are fairly sure that you are going to need to make the disk a Startup disk.

5.2 Copying disks

>DISKCOPY> The process of copying a disk onto another makes the second disk a 'photocopy' of the first. Immediately after the copy has been made, the two disks are identical in every way; any processing you do of either disk will stop them from remaining identical.

The command used is DISKCOPY

The only information that DISKCOPY needs is which drive will hold the disk you want to copy (the Source drive) and which drive will hold the disk the copy will be made on (the Target drive). The target drive doesn't have to be a different drive from the source drive, but if you have two drives this is usually the most convenient way of working. If you use the same drive for both source and target, DISKCOPY will prompt you when you need to change the disk in the drive from the source disk to the target disk.

Note: If you want to put the copy on a new blank disk, you don't have to format the new disk before you start to make your copy. DISKCOPY will give the new disk the same format as the source disk.

The steps to copy a disk are as follows:

1 **Display a C> system prompt (or an A> if you have a floppy disk PC)**

If you are not sure how to do this, reset your PC by holding down the **[Ctrl]** and **[Alt]** keys and pressing **[Del]**.

If you have a floppy disk PC, put the MS-DOS Disk in Drive A (the lefthand drive if you have two), and hold down the **[Ctrl]** and **[Alt]** keys and press **[Del]**.

2 **If you have a floppy disk PC place your MS-DOS Startup disk in Drive A (the lefthand drive if you have two).**

This disk will already be in the drive if you have just switched on or reset the machine.

3 **Make the root directory the current directory**

If you have just switched on or reset your PC, you need not do anything. Otherwise you may need to type:

```
CD\ [ ↵ ]
```

To return to working on the root directory.

* 4 **Type:**

```
DISKCOPY A: B: [ ↵ ]
```

What happens next depends on whether your PC has one or two floppy disk drives.

● **If you have a single-drive PC**

After a short while, you will see the following message on the screen:

```
Insert SOURCE diskette in drive A:
Press any key when ready...
```

5 **Remove the MS-DOS disk from the drive and insert the disk you want to copy.**

Press the drive release button to release the disk, then withdraw it completely from the drive. Insert the disk you want to copy into the drive until it latches into place.

6 **Press a character key**

The [↵] key or the Space Bar are both character keys and one of these is usually the most convenient choice.

After a short while, you will see the message:

```
Copying - 80 tracks

9 Sectors/track, 2 side(s)
```

followed by:

```
Insert TARGET diskette in drive A:
Press any key when ready...
```

7 **Remove the disk from the drive and insert the disk you want to store the copy on**

Press the drive release button to release the disk, then withdraw it completely from the drive. Insert the disk, on which you want to store the copy, into the drive until it latches into place.

8 **Press a character key**

The [↵] key or the Space Bar is usually the most convenient choice.

If you are storing the copy on a new blank disk or on a disk that has a different format to your source disk, you will see the message:

```
Formatting while copying
```

You may then see the message:

```
Insert SOURCE diskette in drive A:
Press any key when ready...
```

This means that all the contents of your source disk could not fit into the computers memory at one time. Go back to step 4 * and repeat the process until you see the message:

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

9 **Type [Y] if you want to copy another disk; type [N] to leave DISKCOPY**

● **If you have a two-drive PC**

After a short while, you will see the following message on the screen:

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

5 **Remove the disks currently in the drives and insert the disk you want to copy in drive A (your lefthand drive) and the disk you want to store the copy on in drive B (your righthand drive).**

To remove a disk, press the drive release button and then withdraw the disk completely from the drive. Insert the disk you want to copy and the disk you want to store the copy on, into their appropriate drives until they latch into place.

6 **Press a character key**

The [↵] key or the Space Bar is usually the most convenient choice

After a short while, you will see the message:

```
Copying - 80 tracks
9 sectors/track, 2 sides(s)
```

If you are storing the copy on a new blank disk or on a disk that has a different format to your source disk, you will see the message:

```
Formatting while copying
```

Finally the message:

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

7 **Type [Y] if you want to copy another disk; type [N] to leave DISKCOPY.**

5.3 Checking disks

>CHKDSK> When a program fails unexpectedly, with MS-DOS reporting an error in reading or writing a disk, it may be that your disk has become physically damaged (for example, scratched) but it may be that your program or its data has become corrupted electronically.

The command to use to check a disk for errors is CHKDSK. This will check a disk for :

- errors in the directory structure that would prevent you from accessing certain parts of the directory tree
- errors in the recorded list of disk areas which are used to store a file
- bad directories

Any errors it finds are reported through error messages.

Errors such as these will stop your program from working. CHKDSK can in some cases ease the problem by 'correcting' the error - though perhaps with the loss of some data from a file. However, it will only do this if you specifically select one of the CHKDSK options: its normal action is just to produce a status report on the disk - plus, if you want, a report on a specific file on that disk. The report on a file will tell you, in particular, how many separate ('non-contiguous') areas of the disk are used to store the file. (Re-using areas of the disk freed by deleting a file usually leads to the new files being stored in a number of separate areas of the disk).

Checking how much storage space there is on a disk

1 **Insert the disk you want to examine in Drive A (your lefthand drive if you have two)**

2 **Type:**

```
CHKDSK A: [ ↵ ]
```

If you have a floppy disk PC, insert your MS-DOS disk and type:

```
CHKDSK B: [ ↵ ]
```

If you have only one floppy disk drive, MS-DOS will display the following message:

```
Insert a diskette for drive B:
and strike any key when ready...
```

Press either the space bar or the return key

After a little while, MS-DOS will display a report on the disk, which tells you (among other things) the number of bytes of storage space available on the disk. (1000 bytes holds about 1000 characters). In the case of the hard disk, this process may take a few minutes if there are a lot of files on the disk.

5.4 Comparing disks

>DISKCOMP> You may occasionally want to compare your security copy of a working disk (created by copying the disk using DISKCOPY) with the original disk. For example, you may keep two security copies of a working disk, to which you copy the working disk alternately; you may at some time become confused as to which is the up to date copy.

The command to use is DISKCOMP. This carries out a track-by-track comparison of the two disks and when DISKCOMP finds a difference between the disks, it puts up a message specifiying on which side of the disk and in which track it found the discrepancy. Either one side or both sides of the disk can be compared in this way.

Full details of this command are included in chapter 8.

Note: The DISKCOMP command will only make a valid comparison if the copy was made using DISKCOPY . If you made the back up disk by COPYing each file individually, the comparison will probably fail because although the data is the same and the directory tree is the same, the files themselves are likely to be in different places on the disk.

6. ORGANISING YOUR WORK

This chapter describes how to make and keep your files readily available for use by:

- **Putting your files into groups**

  As you increase the number of files on a disk, you will find that they become much more manageable and more convenient to use if related files are grouped together. MS-DOS then lets you work on just a group at a time.

  Section 6.1 of this chapter explains:

  - Directories (groups of files)
  - Displaying the current pattern of directories
  - Directory names
  - Specifying a path (moving from one directory to another)
  - Starting new directories
  - Getting rid of directories you no longer need

- **Disk housekeeping**

  The selection of files you have on your disks and the way these files are grouped will frequently become out of date. You may well have files you no longer need taking up valuable space on your disk.

  Section 6.2 explains

  - Filenames
  - Using Wildcards to specify a number of files
  - Copying files
  - Transferring files to and from Input and Output Devices
  - Combining files
  - Examining text files
  - Deleting files
  - Renaming files
  - Comparing files
  - Finding out how large a file is
  - Finding out how much room there is on a disk
  - Finding which disk is on a drive
  - Using the Editor to create or change a text file

- **Protecting important files**

  Files are readily erased or overwritten. Section 6.3 describes how to protect your most valuable files - programs you have bought, last year's accounts, etc. - against being erased or overwritten accidentally.

6.1 Putting files into groups: Directories

6.1.1 What is a directory?

On DOS media, a list of files which are grouped together is called a directory.

Programs and data are in files, both in the computer's memory and on disks. Each file has a name to allow it to be identified.

The way computer files are organised when they are stored on a disk, is most easily understood by comparing your disks with a filing cabinet.

Each computer file is like an individual letter or a document in the filing cabinet.

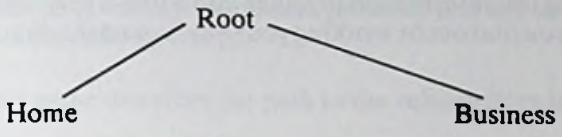
In your filing cabinet, you group together files which are related in some way. Similarly, if you group together, on your computer, files containing programs and data which are related, your files will be easy to manage. So, directories provide a means of organising files on your disks.

Directories are organised into a 'tree-like' structure. You can move from directory to directory, and you can change the way directories are structured by adding and removing directories.

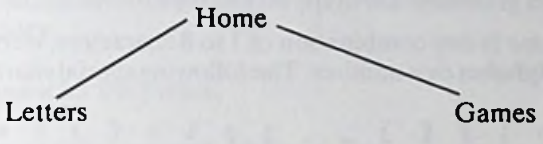
6.1.2 Displaying the current pattern of directories

When you want to find out what directories are used on a particular disk and how these are linked together, you can find this out by working your way around the directories on the disk using CHDIR to move from directory to directory (see Section 7.1) and DIR to tell you what is in each directory. If you start at the Root directory and keep a note of the subdirectories within each directory, you can soon build up a tree structure showing how the directories are linked together.

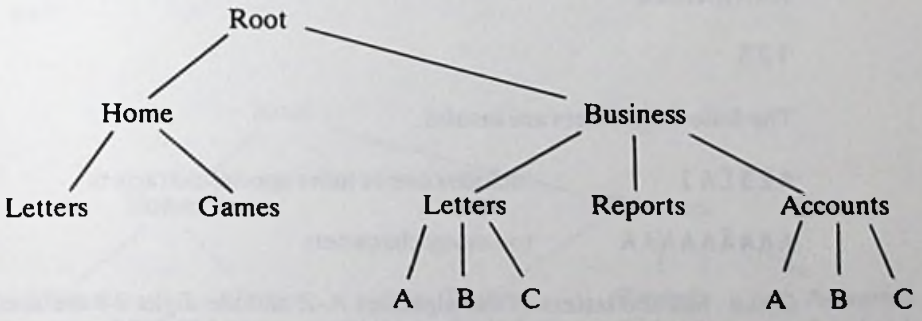
For example, if the Root directory has the subdirectories 'Home' and 'Business', the top part of the tree must be:



If you change directory to the 'Home' subdirectory, you might find this has the subdirectories 'Letters' and 'Games'. Suppose that when you investigate the 'Letters' and 'Games' directories , you found that neither of these directories had any subdirectories. This would give you the lefthand side of the tree as:



Turning to the righthand side of the tree, you might find that the 'Business' directory contains subdirectories called 'Letters', 'Reports' and 'Accounts' and that 'Letters' had subdirectories 'A', 'B' and 'C' directories nor the 'Reports' had any subdirectory, your final directory tree would look like this:



>TREE> Working through the directories in this way is not the only method of discovering what directories there are, and how they fit together. You can also use the TREE command. (See Chapter 8).

The TREE command gives you a list of all the directories on a disk. This list contains path details from the Root directory to each directory listed. It also contains the names of all the subdirectories in this directory. The process of putting this information together into a tree is very similar to creating a tree from what you find out by working your way around the directory structure.

6.1.3 Directory Names

To make your directories more manageable, choose directory names which are relevant to the contents of the directories.

Directories can be given a name which is in two parts - a Name and a Type, although in practice, the type is rarely specified.

The Name is any combination of 1 to 8 characters, starting with one of the letters of the alphabet or a number. The following special characters cannot be used:

```
< > = ¦ / [ ] , . : ; \ + { } ! * ? " ^
```

and you may not include any spaces in the name.

For example, the following names are valid:

```
A

AAAAAAAA

123
```

The following names are invalid:

`A23[4]`	includes one or more special characters
`AAAAAAAAA`	too many characters

Often, just the letters of the alphabet A-Z and the digits 0-9 are used. Note that Lower Case letters (a z) are exactly the same to MS-DOS as upper case letters (i.e. A Z) Cat, for example, is exactly the same as cat or CAT.

Note You must not use the names AUX, COM1, COM2, LPT1, LPT2, LPT3, LST, NUL and PRN.

The Name will usually be chosen to remind the user about the files the directory holds, but this does not have to be so.

The Type is up to three characters long and again the special characters and spaces cannot be used.

A directory can be given any valid name provided it is not already the name of a subdirectory in the same directory.

A subdirectory name describes the path to the subdirectory holding the files you want to use.

6.1.4 Specifying a Path

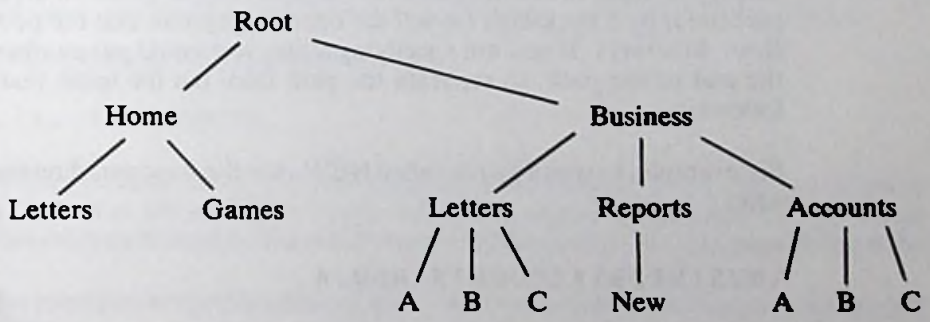
A 'Path' is used to direct MS-DOS to a particular subdirectory on a DOS disk. It is a list of the subdirectories between a point the operating system recognises and the new directory.

The starting point for the path is:

- either the root directory on the drive
- or the directory you are currently working with, on the drive

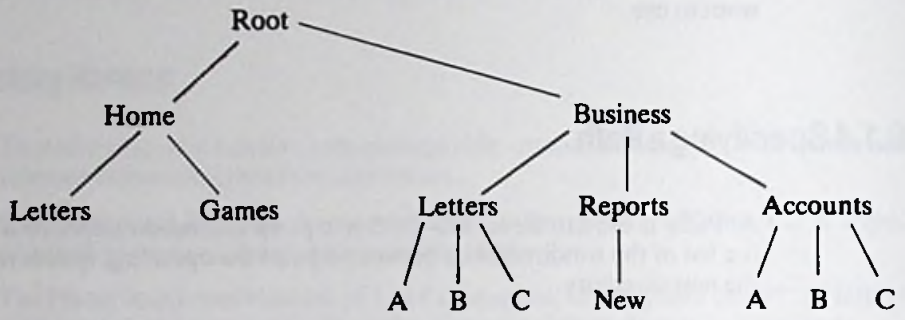
You can use the path from the Root directory regardless of which directory you are currently working with.

Suppose, for example, that you knew that the structure of directories on the disk was:



Paths from the Root directory

To work out the path you need, you start at the Root directory and list all the names on the way to the directory you want (including the name of this directory). The path is these names, written down in order and separated by backslashes (\)



For example, to direct the operating system to the 'Accounts' directory, the names on the way are:

'Business' and 'Accounts'

and so the path is:

```
BUSINESS\ACCOUNTS
```

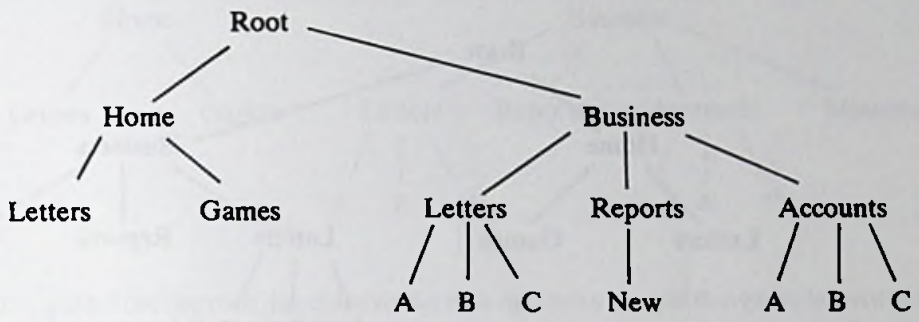
When you come to use this path, for example to specify a file in the directory, you precede it by a backslash (to tell the operating system that the path starts at the Root directory). If you are specifying a file, you would put another backslash at the end of the path, to separate the path from the file name that immediately follows it.

For example, to specify a file called NEW.A in the Accounts directory, you would write:

```
\BUSINESS\ACCOUNTS\NEW.A
```

Paths from the current directory

To work out the path you need, you start at the current directory and, working down the tree as necessary, list all the names on the way to the directory you want (including the name of this directory). The path is these name, written down in order and separated by backslashes (\).



For example, suppose the current directory is 'New'. To direct the operating system to the 'Accounts' directory, the names on the way are:

'Reports', 'Business' and 'Accounts'

(and you could write the path as:)

```
REPORTS\BUSINESS\ACCOUNTS
```

However, 'Reports' is the Parent directory of 'New', and 'Business'is the parent directory of 'Reports', and there is a shorthand way of writing Parent which is..

So the path you would actually write is:

```
..\..\ACCOUNTS
```

When you come to use this path to specify a file in the directory, you would put a backslash at the end of the path, to separate the path from the file name that immediately follows it. **You would not put a backslash at the beginning of the path.**

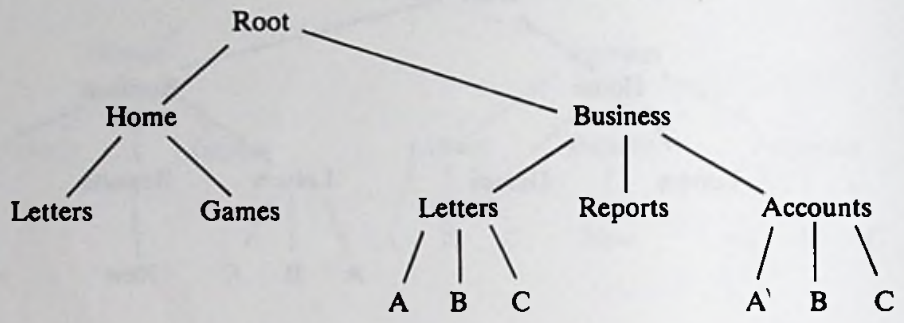
For example, to specify a file called NEW.A in the Accounts directory, you would write:

```
..\..\ACCOUNTS\NEW.A
```

6.1.5 Adding a new directory

>MKDIR> You add a new directory by specifying to MS-DOS where the new directory will fit in. The command used for this is the MKDIR ('Make Directory') command.

For example, if the directory structure you start with is like this:



You might want to add a directory to your existing tree for storing the minutes of meetings. There are two things to define about this new directory - where it fits on the tree and its name.

You define where the new directory is to fit in by specifying the path to it

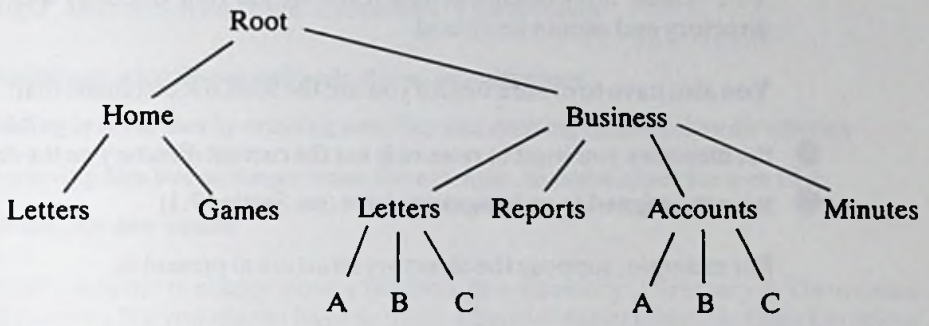
- **either from the root directory**
- **or from the current directory**

The last part of the path gives the name of the new directory; the rest of the path specifies an existing directory which will be the new directory's parent. If the parent directory is to be the root directory, all you will need is a backslash before the name of the new directory; if the current directory is to be the parent directory, you will just need to give the name of the new directory.

You can use any valid name for the directory, provided it is not already the name of a child directory of the same parent.

In this case, we will choose the name MINUTES and make BUSINESS its parent.

New tree:



The path from the root directory to the new directory should therefore be written `\BUSINESS\MINUTES`, so type:

```
MKDIR \BUSINESS\MINUTES
```

When a directory has just been created, it does not have any files in it - these have either to be copied to the new directory from other directories, or created and stored in the new directory. You would have to use the COPY command (see Chapter 8) to copy a number of files to your new directory if, for example, you already had some Minutes files stored in 'Reports'.

A new directory is always empty of file names but if you list the directory you will see it has two entries. These are `\.` and `\..` and they contain the details of the route to the root directory and to the parent directory, respectively. All directories other than the root directory have these entries.

6.1.6 Removing a directory

>RMDIR> A directory is removed from the directory sturcture by using the RMDIR ('Remove Directory') command.

Before you use this command, you have to ensure that

- **there are no files in the directory**
- **any child directory is cleared of files and removed from the directory structure first**

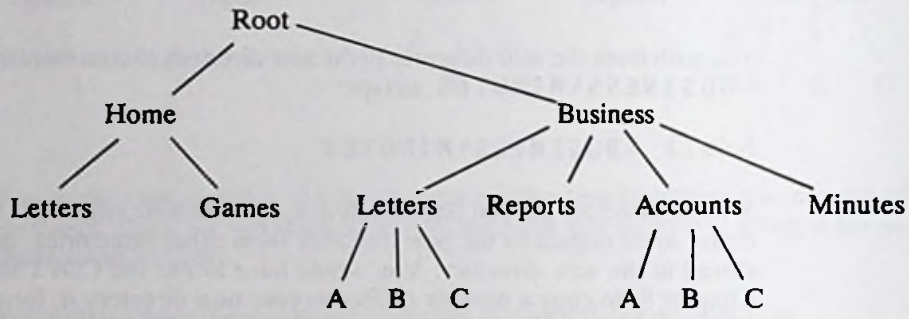
Remember, you can clear a directory with a single DEL command finishing with either a backslash (\) or *.* (see section 6.2.3). But do clear only the directory you mean to clear!

Note: You cannot (and should not attempt to) clear a directory of the entries \. and \.. These hold details of the route to the root directory and of the parent directory and cannot be erased.

You also have to ensure before you use the RMDIR command that:

- **the directory you want to remove is not the current directory on the drive**
- **it is not assigned to an imaginary drive (see Section 7.1)**

For example, suppose the directory structure at present is:



and you wanted to remove Accounts from your directory structure.

You would first have to remove Accounts A, Accounts B and Accounts C. That means you would have to

- first, erase all the files in these three directories, and
- then, use the RMDIR command to delete the directories themselves

Only after that can you remove the Accounts directory itself.

If you made 'Accounts' the current directory when you were removing the directories 'Accounts A', 'Accounts B' and 'Accounts C', you will now need to change the current directory to some other directory, such as 'Business', before you remove 'Accounts' itself.

6.2 DISK HOUSEKEEPING

Disk housekeeping is about keeping your disks organised, with your files on the 'right' disks and in the 'right' directories. This involves

- **finding out what files are already stored in a directory**
- **adding to these files by creating new files and copying files in other directories**
- **removing files you no longer want, for example, to make space for new files**
- **giving files new names**

It isn't possible to simply move a file from one directory (Directory A) to another (Directory B): you always have to make a copy of the original file from Directory A.

This section looks at the commands needed to find out about files already stored in a directory, make copies of files, erase files and rename files. Most new files are created by the programs you run but occasionally you will need to create additional files yourself. This is best done with the help of the PC's text editor, RPED (see 6.2.13). Alternatively, you might use MS-DOS's text editor EDLIN described in Chapter 8. If you are sure you can type the file without mistakes, you can create the file by directly copying what you type into a file. Details of this are given in Section 6.2.3.

To use the commands described in this chapter, effectively, you need to understand about DOS directories and paths. If you are not certain that you do, read Section 6.1 first. For information on how to make Hard Disk backups refer to Section 7.1.

6.2.1 About filenames

The rules about what you can call a file are laid down by a computer's operating system. Files are usually identified by a two-part name. The first part is called the Filename and the second part is called the filetype. In some cases, the file has just a filename and no filetype.

Confusingly, people often refer to both the filename and the filetype together as the filename. This is never done in the manual; the term 'file name' may be used to represent both the filename and the filetype.

The filename is any combination of 1 to 8 characters, starting with one of the letters of the alphabet or a number. The following special characters cannot be used:

```
< > = ¦ / [ ] , . : ; \ + { } ! * ? " ^
```

and you may not put spaces into a filename.

For example, the following filenames are valid:

```
A
AAAAAAAA
123
```

The following filenames are invalid:

`A23[4]`	includes one or more special characters
`AAAAAAAAA`	too many characters

Often just the letters of the alphabet A...Z and the digits 0...9 are used. As far as filenames go, lower case letters (i.e. a...z) are exactly the same to MS-DOS as upper case letters (i.e. A...Z). For example, the filename `Cat` is identical to the filename `cat` or `CAT`.

Note: The filenames `AUX`, `CON`, `COM1`, `COM2`, `LPT1`, `LPT2`, `LPT3`, `LST`, `NUL` and `PRN` have special meanings and should not be used for your own files.

The filename will usually be chosen to remind you about the information the file holds but this doesn't have to be so.

The filetype is up to three characters long and again the special characters and spaces cannot be used. But unlike the filename, the filetype can be fixed by the information the file holds. For example:

- files containing programs which run directly from the MS-DOS operating system must have the filetype COM or EXE
- files containing programs set up for working with the WINDOWS software have the filetype PIF

Always check in your program's user guide to see if the data files associated with this program must have a particular filetype. If no filetype is specified, choose one yourself.

Any valid name can be used for a file provided the same name hasn't already been used for another file in the same directory.

When you want to specify a particular file, you write its name as *filename.filetype*: for example, if the filename is MYFILE and the *filetype* is TXT, you would write the complete name as MYFILE.TXT (You must always remember to put the full stop between the filename and the filetype).

6.2.2 Using Wildcards to specify a number of files

Some programs are able to work on a number of files, one after another. You can make use of this to save yourself typing a separate instruction for each file provided you can construct a 'template' that at least some of your chosen files will match.

The 'tools' you can use in constructing this template are two special characters called wildcards because they can represent any valid character. The two wildcards are:

? which represents one valid character (or blank) in this place in the name

* which represents any number of valid characters (including blank) finishing off this part of the file's name

For example, suppose a directory contained the following files:

A.COM, AB.COM, ABB.COM, B.COM, A.BAT and B.BAT

?.COM would be the template for A.COM and B.COM

*.COM would be the template for A.COM, AB.COM, ABB.COM and B.COM

Both ?.BAT and *.BAT could be the template for A.BAT and B.BAT

A?.COM would be the template for A.COM, and AB.COM

A*.COM would be the template for A.COM, AB.COM and ABB.COM

A.* would be the template for A.COM and A.BAT

?.* would be the template for A.COM, B.COM, A.BAT and B.BAT

. would be the template for all the files

Note the use of a separate asterisk for the filename and the filetype parts of the file's name.

6.2.3 Making copies of files

>COPY> You make copies of files to:
>XCOPY>

- **give yourself reserve or 'back up' copies of the files to use when the original file is damaged or accidentally destroyed**
- **move one or more files from one directory to another (delete the original file once the copy has been made)**
- **create a new file that combines a number of existing files**
- **and, because MS-DOS sees its input and output devices as files, to transfer a file you create at the keyboard into a disk file, output a copy of a disk file on a printer, or transfer what you type at the keyboard to the printer**

The original files are left unchanged by the process.

The MS-DOS commands used to make copies are COPY and XCOPY, full details of which are given in Chapter 8. XCOPY will only handle disk files and cannot combine files in the way COPY can, but it can search a directory's subdirectories for the files you want to copy - something that COPY cannot do.

Copying a file makes a new file that contains all the same information as the original. This copy can be stored either in the same directory as the original or in a different directory on a different disk.

The methods of copying described here, make copies of one or more files. However, the files must all be in the same directory, so if you want to copy files from separate directories you must repeat the process for each directory in turn.

The originals of the files are known as the 'Source' and where the copies are stored is known as the 'Destination'.

The steps are as follows:

1 **Decide on the file or files you want to copy. These are described as your source files.**

You can use one command line to copy a number of files provided these can be described by the same file template (i.e. a file name covering a number of files by including the wildcard characters * and ?) and provided you do not want to store the copies in the same directory as the original files. If you want to make duplicate copies in the same directory you have to copy the files one at a time.

2 **Jot down the drive, directory and file name (or template) you are going to use for the source file(s). These make up the source details for the copy.**

Write this down in the form *drive:\directory\filename*. For example, if you want to copy all the files with the filename MYFILE in a directory called SUBDIR on Drive A, you would write:

```
A:\SUBDIR\MYFILE.*
```
if SUBDIR is listed in the Root (principal) directory on Drive A.

```
A:\DIR1\SUBDIR\MYFILE.*
```
if SUBDIR is listed in the DIR1 directory, which is itself listed in the Root directory on Drive A.

3 **Decide on where you want to store the new files and what they are to be called. This is known as the destination of the copies.**

Often you will want to store the copy in a different directory but under the same filename and filetype. Make sure that you do not already have any files with this filename and filetype already in your destination directory or, if you do, make sure that you no longer need the files.

It is also wise to check that there will be enough room for the new file on the destination disk - see Section 4.7.

4 **Jot down the drive, directory and file name (or template) you are going to use for the destination file(s). These make up the destination details for the copy.**

Write this down in the form:

drive:\directory

if you want to use the file name(s) that the source file(s) already have

```
drive:\directory\filename
```
if you want to use different file name(s)

5 **Type in the appropriate COPY command**

The command you need will be:

COPY *source-details destination-details* [↵]

MS-DOS then makes the copy or (copies) you have requested.

6.2.4 Transferring files to and from Input and Output Devices

Transferring information

- **from an Input Device such as the keyboard into a disk file**
- **from a disk file to an Output Device such as the display or the printer**
- **from an Input Device to an Output Device**

are all copy operations. They use COPY commands very like those used to copy disk files between different drives, but with an Input Device as the Source for the copy and/or an Output Device as the Destination.

The way to tell MS-DOS that, for example, the keyboard is the source and the printer is the destination, simply put its device name in the relevant position in the COPY command. The keyboard's name is CON and the printer's name is LPT1, and so the command line you would need in this instance would be:

```
COPY CON LPT1
```

Once you have given this command, everything you type in at the keyboard will simply be printed out on the printer until you type Ctrl-Z.

Similarly, you could tell MS-DOS to transfer information it receives through the Serial Interface (known to your computer as AUX) to the disk file called TRANSFER.DAT with the command line:

```
COPY AUX TRANSFER.DAT
```

The device names associated with the PC's input and output devices are listed in Part 1.

6.2.5 Combining a number of files into one large file

The COPY command also allows the data from a number of sources - disk files, the keyboard, the serial interface - to be combined into a single file in one copying process. The files that are combined into a single file in one copying process. The files that are combined do not have to be stored onto a disk: one of the files could be input from the keyboard and another could be input from the serial interface.

The resulting file can either be a new file or the first of the source files, with the information from the other files appended to the end of this file. The other source files are not affected by the copying process.

Details of the commands that combine files in this way are given in the full description of the COPY command in Chapter 8.

6.2.6 Examining text files

>TYPE>
>PRINT>

Many files can only be interpreted by the program that uses them and so using the file is often the only way of checking up on what the file contains. However, text files - that is, files of ASCII characters that are entirely free of special program codes - can readily be displayed on your screen or printed out on a line printer. Trying to display non-text files in this way will only result in output that is impossible to read.

The commands used to display text files are TYPE and PRINT. The TYPE command is used to display the text file on your PC's screen; the PRINT command is used to output the file on a printer attached to your PC.

The TYPE command can display only one file at a time and unless you make the display pause, for example by using the MORE filter (described in Section 7.3), much of the file will be scrolled off the screen before you have time to read it. However, it provides a simple and quick way of checking whether a particular text file contains what you expect.

The PRINT command can be used to print either one or a number of files. It sets up a list of files that are to be printed and sends these to the printer in the background while you get on with doing other jobs.

You cannot change any of the files that are waiting to be printed and you must not take the disk they are stored on out of the drive until all the files have been printed.

However, you can change the list after the files have started being printed. You can:

- **add new names to the end of the list**
- **remove the names of files you no longer want to print**
- **abandon the rest of the print list**

How this is done is covered in Chapter 8, where the PRINT command is covered in detail.

6.2.7 Deleting files

>DEL>
>ERASE>

Unwanted files will clutter up your disk space unless you delete them. Deleting a file reclaims the space the file occupied both in the directory and on the disk, freeing it for use for some other file. Files are not deleted if:

- they are not in the current directory of the disk
- they are write-protected

The command to use is DEL or ERASE. ERASE is the alternative name for the DEL command.

Once a file is deleted, it cannot be recovered, so it is important that only files you do not want are deleted. The way to ensure this is to write-protect all important files (see Section 6.3)

This is especially important if you plan to delete a number of files by giving a file template constructed using the wildcards * and ? (see Section 6.2.2). MS-DOS only asks you if you are sure you want to delete every file that matches the template, if you ask to delete all the files in a directory. In all other cases, you could discover that your template matched a file you did not mean to delete.

The steps are as follows:

1 **Decide on the file or files you want to delete**

You can use one command line to delete a number of files provided these can be described by the same file template i.e. a file name covering a number of files by including the wildcard characters * and ? (see Section 6.2.2).

2 **Jot down the drive, directory and file name (or template) you are going to use. These make up your file-details.**

Write this down in the form drive: *directory* filename. For example, if you want to delete all the files with the filename MYFILE in a directory called SUBDIR on Drive A, you would write:

```
A:\SUBDIR\MYFILE.*
```

if SUBDIR is listed in the Root (principal) directory on Drive A

```
A:\DIR1\SUBDIR\MYFILE.*
```

if SUBDIR is listed in the DIR1 directory, which is itself listed in the Root directory on Drive A.

3 **Type in the appropriate DEL command**

The command you need is:

DEL file-details [↵]

MS-DOS then deletes the files you have specified, unless you have asked it to delete all the files in the directory when it first puts up the message:

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

Type **[Y] [** ↵ **]** if you want all the files to be deleted; type **[N] [** ↵ **]** and then re-write your command line if you had not intended to delete all the files.

6.2.8 Renaming files

>RENAME> Files do not have to keep the names they were given when they were created. The names can be changed at any time to any other valid filename by using the RENAME command. However, you must not use a name that has already been given to another file or to a directory within the same directory.

Renaming a file does not affect the contents of the file in any way; it just changes the filename recorded in the directory. Because of this, renaming cannot move a file from one directory to another, even on the same disk: to do that, you have to make a copy in the new directory and then delete the old version.

The name you choose should:

- be different from any other file or subdirectory name in this directory
- suit the needs of the programs you are going to use to process the file (refer to the programs's user guides)
- remind you of the information the file is used to store

The usual characters to use in the name are the letters A...Z and the numbers 0...9.

The steps are as follows:

1 **Jot down the drive, directory and file name of the file you are going to rename. These make up your *file-details.***

Write this down in the form *drive:\directory\file-name*. For example, if you want to rename a file called MYFILE.TXT in a directory called SUBDIR on Drive A, you would write:

```
A:\SUBDIR\MYFILE.TXT
```

if SUBDIR is listed in the Root (principal) directory on Drive A

```
A:\DIR1\SUBDIR\MYFILE.TXT
```

if SUBDIR is listed in the DIR1 directory, which is itself listed in the Root directory on Drive A.

2 **Jot down the new file name you have chosen for the file.**

For example, if you have decided to call the file MYFILE.VI write down MYFILE.VI

3 **Type in the appropriate REN command**

The command you need is:

`REN` *file-details new name* [↵]

MS-DOS then renames the file, unless you have chosen a name that you are already using. In that case, it displays an error message and does not rename the file. If this happens, either delete the file that is causing the problem or choose another name before trying to rename the file again.

6.2.9 Comparing files

>FC> If you have a number of versions of the same file, you may want to find out exactly what the differences are between these versions.

The command to use is the FC command , which will compare files for you either line-by-line or byte-by-byte. You would opt for:

- **line-by-line comparisons to compare versions of a word-processor document or a BASIC program saved in ASCII.**
- **byte-by-byte comparisons to compare versions of a program**

Starting at the beginning of the two files, FC matches each line or byte of one file against the corresponding line or byte of the other file. When FC spots a difference, it carries on comparing until the two files match again: the whole section that is different is then either output on the screen or sent to a file.

Because inserting or deleting lines is a common change to make to documents or BASIC programs, FC tries to 'resynchronise' files it is comparing line-by-line. It will keep reading lines into its buffer from each file until it matches a line from one file with a line from the other - or the buffer is filled up. If the buffer becomes full, FC gives up comparing the files and declares the files 'too different'. No such attempt is made to resynchronise files that are being compared byte-by-byte.

6.2.10 Finding out how much room there is on a disk

As you create, copy, edit and delete files on a disk, you may want to check how much storage space is currently available on the disk. (When you edit a file you typically need at least as much free space on the disk as the size of the file, and it is wise to check this before you start editing the file.)

The easiest way of getting this information is to use the DIR command (see Section 7.1) to display details of a directory or even of a single file on the disk. At the bottom of the display will be quoted the number of bytes free on the disk. (1000 bytes will hold about 1000 characters.)

Compare this number with the size quoted for the file you want to copy onto the disk or the size of the file you want to edit, before deciding to go ahead.

6.2.11 Finding which disk is on a drive

>VOL> Reads the label on the specified drive (see also LABEL command, in Chapter 8).

6.2.12 Finding out how large a file is

1 **Jot down the drive, directory and file name of the file you want to know the size of. These make up the file-details.**

Write this down in the form drive:\directory\file-name. For example, if you want to rename a file called MYFILE.TXT in a directory called SUBDIR on Drive A, you would write:

```
A:\SUBDIR\MYFILE.TXT
```

if SUBDIR is listed in the Root (principal) directory on Drive A

```
A:\DIR\SUBDIR\MYFILE.TXT
```

if SUBDIR is listed in the DIR1 directory, which is itself listed in the Root directory on Drive A

(Turn to Section 6.1.4 if you need extra help in working out what to write down)

2 **TYPE:**

`DIR` *file details* [↵]

Among the details that will appear on the screen will be the size of the file in bytes. (1000 bytes is about 1000 characters.)

6.2.13 Using the Editor to create or change a text file

Additional External Command

The PC is supplied with a screen-based Text Editor called RPED which is very useful for editing small text files. Because it shows the file you are editing on the screen, it is very easy when you are using RPED to look over sections of the text file you are creating or changing, spot a mistake, move the curser to this mistake and make your corrections.

The main use you are likely to make of RPED is for creating or updating files that have the filetype BAT, in accordance with instructions in your programs' user guides. (These guides may suggest you use the MS-DOS text editor EDLIN but you will probably find using RPED easier.) Other files are usually all created by the programs you run and updated by them as well.

The RPED program is stored on the MS-DOS disk.

Overview of using RPED

The first screen that RPED displays tells you about the keys on your keyboard you will be able to use while you are working with RPED. It also asks you whether you want:

- to edit (ie. change) an existing file
- to immediately re-edit the file you have just finished editing
- to create a new file

and it tells you which key to press to make your selection. Put the disk holding the file you want to edit (or the disk you are going to store the new file on) in Drive A and then press the key appropriate to the job you want to do.

RPED then asks you questions about the file you want to edit. If you want to edit an existing file, RPED needs details of where this file is stored. If you want to create a new file, RPED just asks you where the edited version is to be stored

After you enter this information and press [↵], you will see either the current version of the file displayed on the screen or a blank page ready for you to type in your new file. Messages on the screen tell you which keys to press to do such actions as:

- inserting a new line
- deleting a line

Running RPED

1 **Display an A> system prompt.**

Switch on and load MS-DOS, by following the Startup procedure given in Part 1, Chapter 3, and using MS-DOS Disk as the startup disk. If you are already using MS-DOS on your PC, finish what you are doing then return to an A> system prompt and type:

```
CD \ [↵]
```

Note In the case of a Hard Disk PC the RPED program should be already available in the root directory of disk C.

2 **Type:**

```
RPED
```

After a short while, the following should be seen on the screen. This is RPED's opening screen:

```
This screen is for small files (up to 750 lines) and uses
normal Cursor, Page, Home, End, Insert and Delete keys.

f1 = Edit Existing File
f2 = Re-edit Previous File
f3 = Create New File
f4 = Quit
```

6.3 PROTECTING YOUR FILES

As you store programs and data in files, there are a number of questions you should give some thought to:

- **Will you want to maintain a security copy of the file?**
- **Do you want to protect the file against being accidentally deleted or overwritten?**
- **Will you want to know when the file was last changed?**

These questions are all ones of file security and we strongly recommend you to pay attention to this aspect of using your PC. In particular, we would advise you to make security copies of your valuable files. If you do this regularly, you may not need to take any other steps to protect your files.

6.3.1 Making security copies

You will want to keep security copies of most of your files, for use in case of accident. The process of creating such copies is known as 'back-up'.

There are a number of possible ways of making back-up copies of files:

- you can make copies of floppy disks, by using the DISKCOPY command (see Chapter 5) or the XCOPY command (see Section 6.2).
- you can copy individual files onto a different disk, by using the COPY command or the XCOPY command (see Section 6.2).
- you can back up one or more files from one disk to another using the BACKUP command; BACKUP works on hard disks and floppy disks.

6.3.2 Controlling the creation of security copies

If you have a group of files you back up regularly, it saves you time if you only backed up those files in the group that had been changed since the last time you backed the files up. You will want to do this whether you created your security copies, by using a XCOPY command, by using a DISKCOPY command or a number of COPY commands.

You get the required control when using the XCOPY command through its /A and /M options.

6.3.3 Protecting against accidental deletion

>ATTRIB> When a file is created it is a Read-Write file. This means you can:

- read it
- change it
- delete it

This in turn means you can accidentally destroy it!

Although you wouldn't deliberately delete a file you knew you wanted, you might temporarily forget that it contained important information or you might delete it alongside a group of other files with similar filenames.

The way to ensure that your important files are not lost through being accidentally overwritten or deleted is to make them Read-Only files. This says that the files:

- can be read into the computer's memory and processed
- but the versions stored on the disk cannot be overwritten nor can they be erased.

The files are then said to be Write Protected, and they stay write- protected until you make them Read-Write again.

Use the ATTRIB command to make a file Read-Only - and to make it Read-Write again.

6.3.4 Monitoring when files are changed

MS-DOS helps you monitor when your files are changed, by Time and Date Stamping the files to show when the files were last updated. These times are recorded automatically for your DOS files, and can be displayed when the directory is listed, using the DIR command (see Section7.1).

One use of the time and date stamps is to spot how out of date a copy of some file is. If the time and date recorded for the copy is a long time before that of the version you are currently using, the chances are that the copy is not up to date.

You may also want to take advantage of the time and date stamping if you are one of a number of people using the same data files and you want to know whether a particular file has been changed since the last time you used it. The time and date stamps would quickly tell you whether a file had been changed.

If you use time and date stamping you need to keep your PC's internal clock set correctly.

7. RUNNING PROGRAMS

The primary job of your computer is to run programs for you. These programs can be 'application' programs you buy from your computer dealer, programming tools such as programming languages or assemblers, or the MS-DOS commands described in this manual.

You can recognise a file that contains a program you can run, from its filetype:

- an 'application' program you can run will have the filetype COM or EXE
- an MS-DOS command is either built into MS-DOS or stored in a file with the filetype COM

Remember: Internal commands are always available, but External commands can only be used if they are are stored on a disk in one of your drives and MS-DOS is told where to look for them.

In Part 1 of this manual we described the essential steps in running a program on your PC and through an MS-DOS command line:

- selecting a suitable program
- finding out the information you need from the program's own user guide
- finally setting the program running

In this chapter, we look at the additional facilities that MS-DOS provides.

Section 7.1 describes the fundamental process of running a program - locating the program you want to run and the data files you want it to process, making these files available so that the program runs efficiently, and finally typing in the command line that starts the program running.

Section 7.2 describes some shortcuts you can take at the keyboard.

- Editing the previous command line to save typing
- Stopping the program that is being run. Sometimes you can then make it continue where it left off.
- Keeping a record on your printer of what you have typed and the programs you have run
- Extending the number of disks and directories MS-DOS will search automatically to find the program or data file you want

Programs are usually designed assuming that you will type in any additional information and instructions it needs at the keyboard. They are often also set up to send any information they generate to the display screen. Section 7.3 shows you how to

- tell a program to take its input from a file or from a telephone link
- tell a program to send its output to a file, or to a printer or down a telephone link
- link programs together so that the output of one program is the input of the next program in the chain

without changing the programs themselves in any way

It also describes three special programs known as filters, that take the output from the other programs and process it.

Section 7.4 describes how to ease the burden of running the same series of programs repeatedly, by telling MS-DOS the details of the sequence of programs once. On future occasions, you can type in one command line and MS-DOS will run the programs one after another automatically. A special case of this is a series of programs which you can arrange for your PC to run whenever it is turned on.
When you buy programs for your PC to run under the MS-DOS operating system, remember you need to choose ones that are:

- **stored on 3 $^1/_2$ inch floppy disks**
- **designed to run under either the MS-DOS or the PC-DOS operating system**

> VER > In a few cases, you may need to know which version of MS-DOS you are using in order to be sure that the program will run on your PC. You can readily find this out by using the VER command.

Programs supplied with your PC and many commercial programs can be run simply by typing in the appropriate command line and they will work perfectly. Other programs will run but won't always produce quite the results you expect because they don't control the screen properly or they don't respond correctly to what you type. Such programs need to be 'installed' before you use them. How to go about installing a program is described in Part 1.

It is important to know what a directory is to understand the commands covered in this chapter. If you aren't certain you know, read Section 6.1 before you use the commands.

7.1 RUNNING A PROGRAM

Before you run any program, you need to know which disk files you will need, both to start the program running and while it is running. You also need to know how you are going to ensure that these files are found by MS-DOS when they are needed.

Usually (but not always) the program's own user guide will tell you which files you need and your job is simply to ensure that all the relevant files are on disk in one of your drives and that you can tell MS-DOS, through the program, where to look for the files.

This section covers:

- **finding the files you want to use by listing sections of DOS directories**
- **accessing files from different directories**
- **running the program**

Note: (i) If the program you want to use wasn't specifically intended for use on the PC, you may will be able to run the program but the way the results are displayed could be not quite right. Such a program may need installing for use on your PC. How to go about this is described in Part 1, Buying and Installing Programs.

(ii) Always remember to check whether you can make a duplicate copy of the programs you buy, before using them.
If possible, you should always use a copy and keep the original disks safely stored away, to use solely to make further copies as and when these are needed.

(iii) There is advice in Part 1, Buying and Installing Programs, on how to prepare disks holding only the programs and external commands you need while you are using a particular application (for example, your word processor or your spreadsheet).

7.1.1 Finding the files you want

It is always advisable before running any program or using any command to check where the files you want to use are located. The information in a program's user guide, for example, may not always tell you the right directory for a file -particularly if this guide wasn't prepared specifically for your PC. Typically, you will need to find out

- whether a certain group of files is stored in a particular directory
- whether a certain file is stored in a particular directory

You may also want to know details about particular files or groups of files, such as:

- how large are they?
- are they protected in any way?
- when were they last edited?

> DIR > This information is obtained by displaying sections of the directory with the DIR command.

Telling MS-DOS which files you want information about comes in two parts:

- telling MS-DOS which directory you want information from
- specifying the names of the files within the directory that you want information about

You specify the directory by giving the drive it is on and the path to the directory. (How to do this is described in (section 7.1.2 below). If you do not specify a directory, your PC displays information about the default directory.

You specify the files you want by giving the file template that covers all their names. (A file template is a special type of file name that matches a number of possible file names by including the wildcard characters * and ? (see Section 6.2.2). If you do not specify which files you want, you get information about all the files in the directory.

The amount of information that is displayed when you use the DIR command depends on which command option you select. One option is to display just the filenames and filetypes of the files in the directory. This is often the only information you need and it is displayed compactly with detail of five files on each line. If you do not specify command option, MS-DOS gives one line of the display to each file and shows the size of the file in bytes (1024 bytes - which can also be written as 1 kilobyte or 1K or 1Kbyte - is approximately 1000 characters) and when it was last changed. The time information can be useful if, for example, you wanted to check whether you brought a particular file up to date the last time you used your PC.

7.1.2 Using files in a different directory

Files are organised into directories which are themselves organised into a 'tree-like' structure. MS-DOS only searches or stores files in one directory at a time because this is quick and effective. You will rarely want to process files from different directories at the same time.

Unless you tell it otherwise, the directory MS-DOS uses on a particular drive is whichever directory has been designated the Current Directory.

However, this means you need to know how to direct MS-DOS to a different directory:

- to use a particular file from a different directory.
- to change the current directory on the drive, for example if the command or program you want to use can only process files in the current directory.

One of the first things you should find out about a program or command is whether it needs any files to be in a current directory.

The key to using files from a different directory is the 'Path' either from the current directory or the root directory on the drive.

The Path

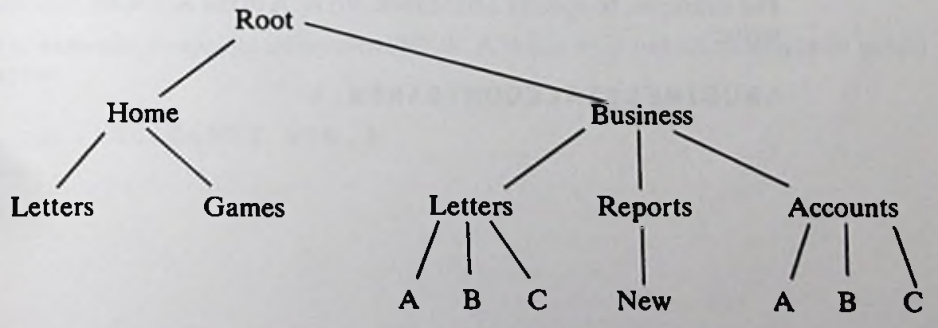
A 'Path' directs MS-DOS to the directory you want to use by giving it a route through the directory tree. The route is essentially just a list of directory names.

The starting point for the path is always:

either the root directory on the drive
or the directory you are currently working with on the drive

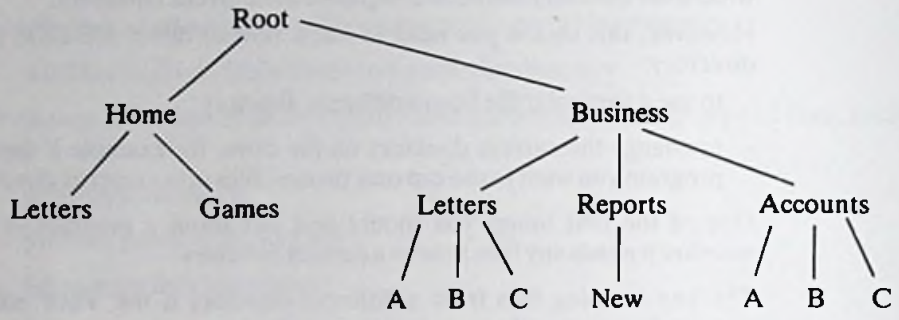
You can use the path from the root directory regardless of which directory you are currently working with.

Suppose, for example, that you knew that the structure of directories on the disk was:



Paths from the Root directory

To work out the path you need, you start at the Root directory and list all the names on the way to the directory you want (including the name of this directory). The path is these names, written down in order and separated by backslashes (\).



For example to direct the operating system to the 'Accounts' directory the names on the way are:

'Business' and 'Accounts'

and so the path is:

```
BUSINESS\ACCOUNTS
```

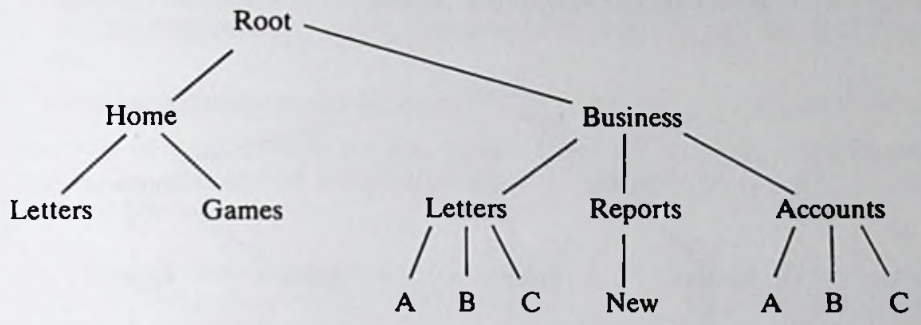
When you come to use this path, for example to specify a file in the directory, precede it by a backslash (to tell the operating system that the path starts at the Root directory). If you are specifying a file, you put another backslash at the end of the path, to separate the path from the file name that immediately follows it.

For example, to specify a file called NEW.A in the Accounts directory, you would write:

```
\BUSINESS\ACCOUNTS\NEW.A
```

Paths from the current directory

To work out the path you need you start at the current directory and working up or down the tree as necessary, list all the names on the way to the directory you want (including the name of this directory). The path is these names, written down in order and sepatated by backslashes (\).



For example, suppose the current directory is 'New'. To direct the operating system to the 'Accounts' directory, the names on the way are:

'Reports', 'Business', and 'Accounts'

However 'Reports' is the Parent directory of 'New' and 'Business' is the Parent directory of 'Reports', and there is a special way of writing Parent which is...

So the path you would write is:

```
..\..ACCOUNTS
```

When you come to use this path to specify a file in the directory, you would put a backslash at the end of the path, to separate the path from the file name that immediately follows it. **You would not put a backslash at the beginning of the path.**

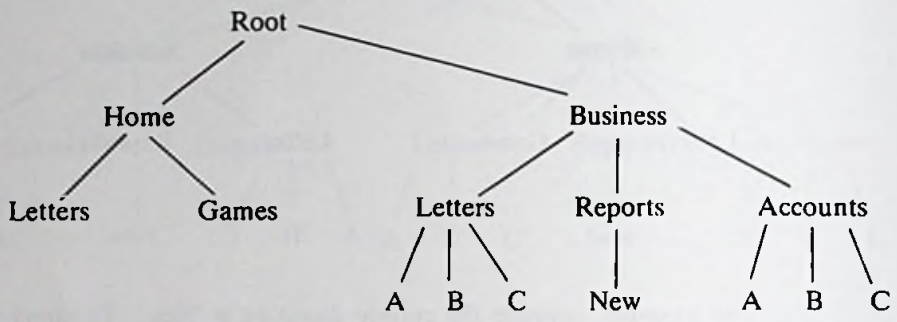
For example, to specify a file called NEW.A in the Accounts directory, you would write:

```
..\..\ACCOUNTS NEW.A
```

Changing the current directory

The current directory on a drive is the directory MS-DOS assumes you want to work on if you do not specify which directory you want to use. It can be any of the directories on the drive. When you first use a drive or when you reset it, the root directory is automatically the current directory.

If the directory structure is like this:



the current directory on this drive might be Business Letters to Company A (ie \BUSINESS\LETTERS\A). If you prepare a new letter, it will automatically be stored with all your earlier letters to this company.

> CHDIR > To change which directory is automatically used on any drive or to see which the current directory is, use the CHDIR ('Change directory') command.

Note: If you want a particular directory to become the default directory - ie, the directory searched when you ask for a file by just its filename - you have to do two things:

(i) Use CHDIR to make this directory the current directory on the drive

(ii) Make the drive the default drive by typing a command line made up of the drive letter for the new default drive followed by a colon (:). For example, to change the default drive to Drive C (the hard disk), you should type:

```
C: [↵]
```

Having more than one current directory on disk

The current directory gives you an efficient way of working on just the files in one directory on a particular drive. However, there may be a few occasions when you would like two or more directories on the same disk to behave as current directories.

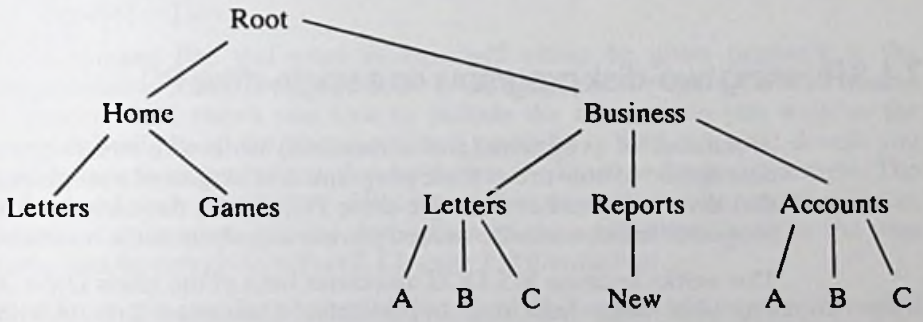
There are two possible sets of circumstances:

- running a suite of programs, when the program files and MS-DOS commands that the programs use, aren't necessarily all in the same directory as the data files
- using commands or programs that only work with files in a current directory

The first of these cases is handled by MS-DOS's file searching facilities (see Section 7.2). The second is handled by treating the directories as if they were on separate drives.

> SUBST > A directory is made to appear on a separate drive by assigning a drive letter to the directory. This is done with the SUBST command. You simply invent a drive that your PC does not already have a meaning for, for example Drive N, Drive O, or Drive P. (The PC initially has meanings only for Drives A..E. See LASTDRIVE command; Part 1, Chapter 3)

For example, suppose the directory structure was like this:



If you wanted to use a program that cannot handle paths to process files from both the 'Accounts' directory and the 'Reports' directory, you would make, an imaginary drive - say, Drive N 'Reports' is then automatically made the current directory on Drive N. When you come to use your program to process, for example, REPORT.85 in the 'Reports' directory, you can now give directions that the program can handle, ie:

```
N:REPORTS.85
```

The directory assigned to the imaginary drive acts as the root directory on that drive. If this dirctory has Child directories, you can use paths on the imaginary drive to work with the child directories just as you would with one of the standard drives, However, you cannot access its parent directory from the imaginary drive.

Coping with any assumptions your program makes

Any program you run may expect the files it needs to be on a particular drive. For example, it may assume that these files will be on your Drive B disk when in fact you have put the files alongside the program on your Drive A disk. It may not always be possible or practical to move the files so that they are on Drive B.

> ASSIGN > The solution, provided none of the files the program wants are on Drive B, is to tell MS-DOS to route any request for a file from Drive B to Drive A. This is done with the ASSIGN command. However, don't use the ASSIGN command too freely: while this assignment is in force, no file on the disk in Drive B can be used because all instructions to look for a file on Drive B will be diverted to Drive A.

> JOIN > Another problem you may meet is that programs and procedures you have established have built into them the assumption that certain files are in a certain directory on a particular disk when in fact there is no longer room for these files on this disk. For example, suppose you need to move all the files in the 'Business' directory onto another disk.

Rather than change your programs and procedures, you can instead just leave an empty 'Business' directory on this disk and then tell MS-DOS that whenever you ask for files from this directory, it should look for them in a different drive. The command that sets this up for you is the JOIN command.

7.1.3 Running two-disk programs on a single-drive PC

A number of programs (and commands) work with two floppy disks and the description of how to use these programs is often geared towards systems with two disk drives. If you have a single-drive PC, you are not excluded from using these programs and commands: indeed you use exactly the same command line.

This works because MS-DOS associates both of the labels Drive A and Drive B with your single disk drive. In particular, it associates Drive A with one disk and Drive B with a different disk. When MS-DOS needs the Drive A disk, it puts a message on the screen asking you to release the current disk and insert the 'Drive A' disk in the drive: when it needs the Drive B disk, it puts up a message asking you to release the current disk and insert the 'Drive B' disk in the drive. If you remember which disk is associated with which drive, you can't go wrong.

What may surprise you is that MS-DOS will in some circumstances ask, for example, for the disk for Drive A when you know that this disk is already in the drive. This happens because MS-DOS last used the drive as Drive B and it doesn't know that the disk in the drive isn't the Drive B disk. Just press a character key (eg. the Space bar).

Note: The system prompt tells you the current default drive, It does not tell you which drive MS-DOS currently regards your floppy disk drive as.

7.1.4 Running the program

Once you have:

- the disks you need in your disk drive(s)
- the current directories you need on all your drives
- the paths to any other directories worked out
- a fresh MS-DOS system prompt on the screen (with the cursor flashing beside it)

you are ready to run your program

> SET > A few programs may additionally require certain parameters to be set up in advance, which are used to specify the Environment the program works in. For example, the program may work differently depending on the current size of your PC's RAM disk. If this is the case, full details should be given in the program's own user guide. Some of these parameters are set up by specific MS-DOS commands: the others are set by using the SET command.

Programs are run by typing in a command line made up of the command name (the name of the program you want to run) and the command tail (ie. the other details the program needs so that it carries out the job you want), and then pressing the [↵] key.

The command line you need to type will either be given precisely in the documentation about the program or it will be given through a **Form** or **Syntax** statement which shows you how to include the information you want in the command tail. Read the documentation carefully to find out what details you need to type in instead of the placeholders in the Form or Syntax statement. The meaning of each placeholder should be either explained alongside the Syntax statement or in some general description of the conventions used in the user guide, like the one given in Part 2, Chapter 3 of this manual.

In addition, you may have to include details of the location of the program itself if this is neither built into MS-DOS or in the current directory on the default drive (the drive with the letter shown in the current system prompt) or in a directory automatically searched by MS-DOS (see Section7.2.4). You will need to precede the command name with:

[d:][\][path]

included if the program is not in a current directory
included if the path starts at the root directory of the drive
included if the program is not in the default drive

For your program to work, the command line has to be exactly right so take care in typing this in. If you make a typing mistake, use the [←] key to rub out characters and then type the line in again. When you are sure the line is as you want, press the [↵] key.

While the program is running, you may be able to move the mouse instead of pressing the cursor keys and to use the mouse buttons to produce certain effects -depending on the program you are running. If you are interested in making use of this (or you want to turn this feature off), turn to Part 1, 'Buying and Installing Programs', where further details are given.

If your program fails in some way, see if you can work out what has happened and how to stop it failing again, either from the program's own user guide or from Chapter 10 of this manual. Chapter 10 covers troubleshooting a wide range of problems. If this does not solve your problems for you, seek the advice of your dealer.

Note:
>CLS>
>VERIFY>

There are two other MS-DOS commands you may find helpful when you are running application programs:

`CLS`	clears the screen and puts a fresh system prompt in the top lefthand corner
`VERIFY`	instructs MS-DOS to confirm every time it writes data to a disk, that this data can be read back, for example, by checking that there are no bad sectors.

7.2 SHORTCUTS AT THE KEYBOARD

7.2.1 Editing the command line

When you make a mistake or change your mind about the command or program you want to run before you press the [↵] key, you can always use the **[Del]** key to rub out characters and then type in the rest of the line again.

Once your command has failed because of some small error, you might think that you have to type in the whole command line again. However, MS-DOS automatically stores your command line and, by pressing certain keys, you can build parts of the stored line into a new command line.

As you do this, you need to imagine a cursor moving through the stored command line dividing the command line into a section you have 'used' and a section that still remains for you to use. This cursor moves 'forward' through the line:

- every time you copy characters from the stored line into your new line
- every time you tell MS-DOS to skip over characters in the stored line
- every time you type fresh characters without telling MS-DOS that you want to insert them.

It also moves 'backwards' if you delete any characters from the line.

The keys you want are as follows:

[F1] or [→]	Add the next character from the stored line.
[F2] *character* (ie. press **[F2]** and then a character key)	Add everything in the stored command line between the current position of the cursor and the given character. The character itself is not added to the new line.
[F3]	Add all the characters in the stored command line to the right of the cursor.
[F4] *character* (ie. press **[F4]** and then a character key)	Delete every thing in the stored command line to the left of this character.

[Del]	Delete the next character from the stored line.
[Ins]	Insert the following characters into the new command line without moving the cursor through the stored command line. If you don't press the **[Ins]** key, this cursor will move forward one position for each character you type. Press **[Ins]** again after you have typed the last character you want to insert.
[←]	Delete from the new command line the character to the left of the cursor. The cursor in the stored line also moves back one character.
[Esc]	Abandon the current new command line and start again working with the old stored line. MS-DOS places a \ at the end of the line you have abandoned and gives you a new line to work on.
[F5]	Replace the stored command line with the command line you have created so far and then start again working with this new line. MS-DOS places a @ at the end of the line you have stored and gives you a new line to work on. (For use when you notice that you made a mistake back near the beginning of the new command line but, naturally, you want to save your changes since then.)
[↵]	Send the new line MS-DOS for processing. It also replaces the stored command line with this new line.

For example, suppose that your last command was:

```
COPY MYFILEB:\DIR1\FILE
```

and you wanted the next command to be:

```
COPY MYFILEC:\DIR2\YOURFILE
```

The first character you want to change is B, so press **[F2]** and then type B. On your screen will appear everything up to but not including the B ie.:

```
COPY MYFILE
```

In the stored command line, the cursor moves up to the letter B

Type C. That gives you on the screen:

```
COPY MYFILE C
```

In the stored command line, the cursor moves past the B.

The next character you want to change is 1, so press **[F2]** and then type 1. Once again, everything from the current position of the cursor in the stored line up to but not including the character you typed is shown on the screen, giving you:

```
COPY MYFILE C:\DIR
```

Type 2, giving you:

```
COPY MYFILE C:\DIR2
```

Now you want just the next character from the old line, so press **[F1]**. Your new command line is now:

```
COPY MYFILE C:\DIR2\
```

Next you want to insert YOUR, without moving the cursor through the stored command line - so press **[Ins]**, type YOUR and then press **[Ins]** again. This gives you:

```
COPY MYFILE C:\DIR2\YOUR
```

The final stage is to add the remainder of the stored line (FILE) to the new line, which you do by pressing **[F3]**. Your new command line should now be:

```
COPY MYFILE C:\DIR2\YOURFILE
```

and you will have got this line with a total of 14 keystrokes instead of 28.

If you had forgotten to press **[Ins]** before typing YOUR, you would have effectively overwritten the FILE in the stored command and after pressing **[F3]**, you would still have:

```
COPY MYFILE C:\DIR2\YOUR
```

7.2.2 Stopping a program while it is running

From time to time, you may start a program running and then quickly realise that you don't want to run this program after all. You will be able to stop most programs from the keyboard by entering Ctrl-Break ie. holding down the **[Ctrl]** key and pressing the **[Break]** key.

> BREAK > The length of time it takes before the program actually stops depends on the type of work the program is doing. If it is not taking input from the keyboard or sending output to the screen or to a printer, it may be a while before the program stops. If the time the program takes to stop is too long, you can use the BREAK command to add to the number of occasions MS-DOS looks at the keyboard to see if Ctrl-C or Ctrl-Break has been entered. This is particularly useful while you are developing a program that uses your disks very much more than the keyboard, the screen or the printer.

Some programs are designed to allow you to halt the program temporarily by pressing a particular key or combination of keys. A similar combination of keys may allow you to restart the program - for example, after you have noted down information that is being written on the screen. One such program is the TYPE command (see Section 6.2) which allows you to halt it by entering Ctrl-S and then to restart it by entering Ctrl-Q. You will need to study the documentation provided with your program to find out if you can use keystrokes like these to halt and restart it.

7.2.3 Keeping a record of the work you do

If you have a printer attached to your PC you can use this to record everything that appears on the screen i.e. everything you type in and everything your programs display on the screen for you. You could find recording all your actions a great help when it comes to analysing the work you have done or in preparing for future work using the same programs.

To start recording on the printer all the instructions you type in and all the messages MS-DOS displays enter Ctrl-P. Ctrl-P stops this information being output on the printer.

However, do not expect the print-out always to be identical to what you see on the screen. It will only be the same if the program writes to the screen line-by-line: spreadsheets, for example, do not.

> GRAPHICS >
[Prt Sc]

Another way of recording what is displayed on the screen is to print an image of the screen. This is done simply by pressing a special combination of keys on the keyboard, but if there are pictures on the screen (rather than just text), you first need to prepare your PC by using the GRAPHICS command .

Note: You will normally be able only to print low resolution two color pictures.

Once this has been done and the printer is both switched on and on-line, you only need to press **[PrtSc]** each time you want to print the current display. Provided your PC has been properly set up for your printer, a picture of your screen will be printed in a few minutes.

DO NOT PRESS THE [Prt Sc] KEY IF THERE ISN'T A PRINTER ATTACHED TO YOUR PC

7.2.4 Searching for files

A tidy approach to using your PC is to keep:

- MS-DOS external commands in one directory
- the program files and batch files of an application software package in another directory
- the data files associated with running the software package in a third directory

MS-DOS encourages this approach by searching a sequence of directories for files as if these direcrtories were part of the default directory. This means that you can keep program and Batch files in directories other than the default directory but treat them in command lines as if they were in the default directory. (Batch files are described in Section 7.4).

You will find this particularly helpful in using applications that make use of development software (eg. a programming language such as BASIC) as well as its own programs or when you need to use MS-DOS external commands alongside your application programs. Continually moving the files containing MS-DOS software so that these are always in the same directory as the application software, is not recommended or even always possible!

Separate methods are used to tell MS-DOS where to look for the files holding programs you want to execute and where to look for the files you want your program to process.

Extending the search for program files

Note: This applies to the program, command or Batch file you want to execute and to any program that is run as part of the Batch process. It doesn't apply to files that the program processes.

> PATH > If you type in a command line that does not specify where the command or program file is located MS-DOS will always look for it in the current directory of the default drive. You can extend its search to a number of different directories by using the PATH command.

The PATH command gives MS-DOS a list of directories to try after it has failed to find the file in the default directory. MS-DOS will work systematically through these directories until it finds a COM.EXE or BAT file with the right filename or it comes to the end of the list - until, that is, you use the PATH command again (or you reset your PC, which loses all the details of the search path).

The PATH command is also used to display the directories that are searched at present.

> APPEND > You must keep the list of directories to be searched up to date. If you delete a directory on the search path (see Section 6.1), you must redo your complete search path. Otherwise, the search will fail whenever MS-DOS tries to search the directory which no longer exists.

Searching for files to process

> APPEND > If you type in a command line that doesn't specify where the files you want to process are located, MS-DOS will always lookfor these in the current directory of the default drive. You can extend its search to a number of different directories by using the APPEND command. Most - but not all - programs can take advantage of this extra searching. (The full description of a program should tell you whether it can be used with the APPEND command to search extra directories).

The APPEND command gives MS-DOS a list of directories to try after it has failed to find the file you have specified in the default directory. MS-DOS will work systematically through your list of directories until it finds the file (or it comes to the end of a list) - until, that is, you use the APPEND command again (or you reset your PC which loses all the details of the search path).

The APPEND command is also used to display the directories that are searched at present.

Note: As with PATH, you should keep the list of directories up to date or the search will fail.

7.3 REDIRECTING INPUT AND OUTPUT

Programs are usually designed to take their standard input from the keyboard and to send their standard output to the screen. Under MS-DOS however, a program can:

- take the input they require from another input device
- send the screen output to another output device
- take this input from a disk file
- store this output in a disk file

without any part of the program having to be re-written.

This section describes how to tell MS-DOS to take the standard input for the program you want to run, from somewhere other than the keyboard and how to send the standard output it produces, to somewhere other than the screen. It also describes how to 'pipe' the standard output from one program to another for further processing - in particular, how to pipe this output to one or more of the three MS-DOS filters FIND,SORT and MORE. These filters:

- **find all the places a given string of characters is used**
- **sort data into alphanumeric order**
- **organise the output of data to the screen so that it is seen a screenful at a time**

Note: The redirection described here applies only to the command line it is used in. If you want to redirect input and output more permanently, turn to Section 4.1.

7.3.1 Redirecting the standard input

You tell MS-DOS to take a program's console input from a different input device or a disk file by ending the usual command line with the details of the source of the input.

These details are written as follows:

<device-name	if the input is to be taken from a different input device
<[d:][\][path\]filename.filetype	if the input is to be taken from a file

A list of device names is given in Part 1.

For example, if the input for the program MYPROG is to be taken from the serial interface, you might have the command line:

```
C>MYPROG <AUX
```
or
```
C>MYPROG <COM1
```

If, however, it is to be taken from the file RUNDATA.1 on Drive C, you might have the command line:

```
C>MYPROG <C:RUNDATA.1
```

Take care, when you redirect input, that all the information the program needs is input through the different input device or stored in the file. The program will stop if it runs out of input data.

7.3.2 Redirecting the standard output

Tell MS-DOS to send a program's console output to a different output device or a disk file by ending the usual command line with the details of the destination of the output.

These details are written as follows:

>device-name	if the output is to be sent to a different input device
>[d:][\][path\]filename.filetype	if the output is to be sent to a file (replacing any existing file of that name)
>>[d:][\][path\]filename.filetype	if the output is to be appended to an existing file

A list of device names is given in Part one.

For example, if the output from the program MYPROG is to be sent to the printer, you might have the command line:

```
C>MYPROG >PRN
```
or
```
C>MYPROG >LPT1
```

If however, it is to be appended to the file RUNDATA.1 on Drive C, you might have the command line

```
C>MYPROG >C:RUNDATA.1
```

7.3.3 Piping output from one program to another

If you have a pair of programs, with the standard output produced by one, exactly the data required as standard input by the other, then you can use MS-DOS to pipe the information from the first program to the second. The second program of the pair is known as a filter.

You pipe information from one program to another by having both command lines on the same line as follows:

command-line-1 ¦ *command-line-2*

For example, if you had a program called FILTER that processed the output from the DIR command, you might have the combined command line:

```
C>DIR C:¦FILTER >STORE
```

where `DIR C:` is the first command line and `FILTER >STORE` the second

If you have another filter program that can process the standard output from the second program, this output can be piped on to the next program and so on, simply by having the separate command-lines on the same line as follows:

command-line-1 ¦ *command-line-2* ¦ ... ¦ *command-line-n*

Typical tasks you might want a filter program for are:

- **to find all the places a given string of characters is used**
- **to sort data into alphanumeric order**
- **to organise the output of data to the screen so that it is seen a screenful at a time**

> FIND >
> MORE >
> SORT>

These types of processing are provided by the three MS-DOS filters FIND, SORT and MORE: FIND does the job of searching through the output for text strings: SORT sorts the lines of output into alphanumeric order: and MORE arranges that the output is displayed a screenful at a time.

These filters provide a good example of using one filter after another. You might well want to sort into alphanumeric order the output from, say, a DIR command and then ensure that the sorted directory is displayed a screenful at a time. You would do this with a command like:

```
DIR C:¦SORT¦MORE
```

Note: You cannot make use of piping within a Batch process

7.4. SETTING UP A SEQUENCE OF COMMANDS

As you use your PC more you are likely to find yoursef typing in the same sets of commands - for example, each time you use a particular suite of commercial software or back-up your important files. This can be both tedious and error-prone.

> BATCH > It would be better to put all the commands into a file and then just type one command which tells MS-DOS to execute this file. This is known as Batch Processing and the file containing the commands is called a Batch file. This section is about creating and running Batch files. It also describes a special Batch file that can be executed immediately after MS-DOS is loaded.

7.4.1 A Simple Batch file

A batch file is a file containing a sequence of MS-DOS command lines. It always has the filetype BAT.

The simplest form of Batch file - and by far the commonest - is just a list of MS-DOS command lines. So if, for example, running your suite of software means you type the three command lines:

```
CHDIR C:\WS
FUNCTION WS.PFK
WS
```

you create a file with these thee lines as the lines of a file. (Use RPED for this - see Section 6.2.) Note: The usual rules apply to choosing a filename for this file; however, take care not to give it the same filename as any of your program or command files because MS-DOS will always execute a program or command file in preference to a Batch file.

If you called this file RUNWS.BAT you would just use the filename RUNWS to run this particular suite of software MS-DOS will execute the command lines one at a time, in the order they appear in the Batch file. In other words, typing:

```
RUNWS
```

is exactly equivalent to typing:

```
CHDIR C:\WS
FUNCTION WS.PFK
WS
```

Notice that you type RUNWS and RUNWS.BAT

The command lines will be displayed on the screen one by one as your PC executes them.

Note: Only if you fully understand and intend their action should you use any of the characters > < or % in a Batch file (even if contained in a REM line).

7.4.2 Making a Batch file more versatile

You may want your Batch file to handle

- **repeated commands or commands that you only want executed under certain circumstances.**
- **a more appropriate way of logging your PC's progress on the screen.**
- **pausing in the middle of batch process to change disks, for example.**
- **carrying out similar actions but with different sets of program files and data files.**

These facilities are provided by some special Batch Subcommands and Batch file parameters. How to use them is described below, with full details of the Subcommands given in Chapter 8.

Repeated or conditional commands within Batch files

> FOR >
> GOTO >
> IF >

In a typical Batch file, you may have:

- the same program file used in a number of command lines , one after the other
- the same data file(s) used in a number of command lines, one after the other
- command lines that you only want executed if certain conditions are met
- short sequences of command lines that are used over and over again in a loop

Batch processing has the subcommands FOR, GOTO and IF (and IF NOT) to handle repeated or conditional actions.

FOR	is used where essentially the same command line is used repeatedly
GOTO	is used to direct MS-DOS to a different command (marked by a label)
IF	is used to make MS-DOS take different actions, depending on the truth of some statement

You can make a whole series of actions depend on a single test by using IF and GOTO together.

Logging your PC's progress on the screen

>ECHO>
>REM>

Normally while a Batch file is running, each MS-DOS command line is displayed on the screen just before it is executed. This is the default way of showing its progress but there are other possibilities:

- displaying command lines and additional messages
- displaying just some special messages
- no on-screen messages

Which option you choose depends what the individual commands in the Batch file do. For example, you might opt for either no on-screen messages or just some special messages if the commands gradually build up output on the screen. If the commands split up into a number of stages, you might want to use additional messages to show which stage your PC has reached.

The Batch subcommands, ECHO and REM, let you determine what appears on the screen:

ECHO	controls whether the MS-DOS command lines are displayed (it doesn't affect the output from your programs in any way) and gives you one way of displaying special messages
REM	also lets you display special messages

If you want both command lines and special messages displayed, use REM for the additional messages. If you don't want the command lines displayed, use ECHO both to turn off the display of command lines and to generate the special messages. This is because turning off the command line display also stops any REM messages from being displayed. (Note: It also stops the special messages associated with PAUSE commands from being displayed - see 'Pausing in the middle of a Batch process' below.)

Another use for REM commands in a Batch file is simply to space out the other commands and so make the file more readable.

Pausing in the middle of a Batch process

> PAUSE >

If you need to swap disks, for example, in the middle of processing a Batch of commands, you should put a PAUSE subcommand in your Batch file. This subcommand is put between the last command line you want executed before you swap the disks and the first one to use the new disks.

When you run the Batch file your PC will stop after the PAUSE subcommand. You can then change disks or whatever. When you are ready for your PC to continue, press one of the character keys on the keyboard (say, for example, the space bar). The next command in the Batch file will then be executed.

The pause subcommand can include a message which will be put up on the screen just before the program halts. You can use this message to remind you what to do before MS-DOS resumes running the Batch file.

If you remove the disk with the Batch file from its drive so that one command line can be executed, you will have to re-insert it before the following command line in the file can be executed. A message will appear on the screen instructing you to put the disk back in the drive. Another PAUSE command will be needed if you want to swap the disks again.

7.4.3 Using Batch files for more than one job

You can use one Batch file to carry out the same operations but on different files by using dummy parameters in the file. Such a Batch file is called a 'Multi-purpose Batch file'.

The standard set of parameters are the characters `%0...%9`. Each of these parameters stands in for a 'value' which you supply when you run the Batch file. For example, suppose you wanted a Batch file that could be used to copy any file in any directory on the default drive to any other directory, you could use one parameter to represent the file and one to represent the source directory and another to represent the destination directory.

Batch files can also make use of environment strings as parameters. These environment strings and the names used to represent them are made available to each program as it is run. They are mainly set up by using a SET command (see Section 7.1). To make use of an environment string in a Batch file, you simply include its name surrounded by percent characters as a parameter. For example, if you wanted the way your Batch process proceeded to depend on whether your environment parameter DISPLAY was CGA or EGA, You would include the parameter `%DISPLAY%` in an IF subcommand in the Batch file like this:

```
IF "%DISPLAY%"=="CGA" GOTO LABEL
```

Parameters can represent any string of characters in a Batch file command line (other than spaces and Batch subcommands themselves). They can therefore be used to represent:

- a filename
- a directory
- a path
- a command line parameter
- a string of characters to compare in an IF command
- a label
- a part of any of these
- any combination of these

For example, in your Batch file MYBAT.BAT, you might represent the filename of a file you want to copy by % and the filename of the copy by %2. You might therefore have a line in the file like this:

```
COPY %1.TXT B:%2.NEW
```

When you run the Batch process you follow the name of the Batch file by the values you want the parameters to represent. MS-DOS then assigns values to the parameters in order, starting with %0 which is assigned to the drive, directory and filename of the Batch file %1 takes the first value in your list, %2 the second, and so on. So, running the Batch file MYBAT.BAT with the command:

```
A:MYBAT File 1 File2
```

would replace each %0 in MYBAT with `A:MYBAT`, each %1 with `File1` and each %2 with `File2`. So MS-DOS would actually obey the command:

```
COPY File1.TXT B:File2.NEW
```

> SHIFT > The ten dummy parameters you get from %0..%9 is usually plenty, but it is possible to use more if you use the SHIFT subcommand. SHIFT can also be used to apply the same group of actions to an unspecified number of files, one after another. How to do this is described alongside the details of the SHIFT subcommand in Chapter 8.

7.4.4 Interrupting a Batch process

You can interrupt a Batch file while running, by entering either Ctrl-Break or Ctrl-C at the keyboard.

The command currently being executed is immediately abandoned but not the rest of the batch commands. Instead, MS-DOS asks you whether you want to abandon these. If you type **[N]** for No, the remainder of the Batch file is processed normally.

7.4.5 Obeying a Batch file automatically

> AUTOEXEC. BAT > MS-DOS will obey a Batch file automatically when you turn your PC on or reset it, if this file is called AUTOEXEC.BAT and it is stored in the root directory of your Startup disk (ie. the floppy disk or hard disk you use to load MS-DOS from).

Immediately after the operating system software has been loaded into your PC's memory, MS-DOS looks to see if there is a file called AUTOEXEC.BAT in the default drive. If it finds one, the commands it contains are executed before you can use your PC for anything else (though you can always interrupt it by entering Ctrl-Break or Ctrl-C).

AUTOEXEC.BAT cannot be a multi-purpose Batch file.

Batch subcommands are listed in alphabetical order, in Chapter 8.

CALL
ECHO
FOR
GOTO
IF
PAUSE
REM
SHIFT

8. MS-DOS COMMANDS

Introduction

This chapter describes all the MS-DOS commands, supplied with your PC. Each command description is listed alphabetically, according to its name, for example, APPEND, ASSIGN, ATTRIB and so on.

Detailed information about how to use the command; the command lines to type; and examples to follow; is given in each description.

If you have difficulty in using the commands, refer to Chapters 1-3 for explanations about how to type command lines properly; why you might want to use MS-DOS commands; and how to substitute the command placeholders (for example, *destination*, *path*, *source*) with the correct information.

If something goes wrong while you are running a command and a system message appears on the screen, see Chapter 10, "Troubleshooting", for instructions on how to correct the mistake.

Append

External command

`Append` *[d:]\path[;[d:]\path...]*

Set a search path for data files

Normally, if you don't say which directory a file is in, MS-DOS tries to find it in the default directory. The APPEND command sets up a sequence of directories that MS-DOS will search for data files along side the default directory. In effect what happens is that each path given in the APPEND command is used in turn with the data file until the file is found. Each path specifies a separate directory. The default directory is always searched first.

This gives you a neat way of ensuring that the data files you need are accessible without clogging up your current directory.

Form `APPEND` *[d:]\path[;[d:]\path...]*

Notes Each path must start from a root directory.

The directories are searched one by one in the order given in the APPEND command until a file matching the file specification is found or the sequence of directories is exhausted.

Not all programs can take advantage of the search path set up by an APPEND command.

Example

The program you plan to use helps you to prepare letters using information from accounts files you have stored on disk in three separate directories: \CLIENTS1, \CLIENTS2 and \CLIENTS3.

To get MS-DOS to search all three directories, use the command line:

```
APPEND \CLIENTS1;\CLIENTS2;\CLIENTS3
```

assuming that the external command APPEND is in one of the directories MS-DOS automatically searches for program files (see Section 7.2.4) and that the three client directories are all on your default drive. If the directories were on Drive B, your command line would be

```
APPEND B:\CLIENTS1;B:\CLIENTS2;B:\CLIENTS3
```

ASSIGN

External command

`Assign` *requested-drive=searched-drive*

Assigns drive letter to another drive

The ASSIGN command tells MS-DOS to convert instructions to read or write files on one drive into instructions to read or write files on another drive. This lets you use drives other than those your program was set up to use.

The ASSIGN command, used on its own without a command tail, cancels all the current 'assignments'.

- **To set up one or more assignments**

Form `ASSIGN` *d=d[d=d...]*

Searched drive

Requested drive

- **To clear one assignment**

Form `ASSIGN` *d=d*

Requested drive

Requested drive

- **To clear all current assignments**

Form `ASSIGN`

You don't have to type the colon after the drive letter when you specify either the searched drive or the requested drive.

Examples

- You want all requests for Drive B to be directed to Drive C. The command line you need is:

```
ASSIGN B=C
```

(assuming that the external command ASSIGN is stored either in the default directory or in a directory that MS-DOS automatically searches - see Section 7.2.4).

- You have assigned both Drive A and Drive B to Drive C, but now you want to have requests for Drive A going to Drive A. the command line you need is:

```
ASSIGN A=A
```

ATTRIB

External command

ATTRIB *[+R|-R] [+A|-A] [d:][\][path\]filename.filetype*

Set file attributes

The ATTRIB command is used to set a file's Read-Only and Archive attributes. These respectively control:

- whether Read-Write or just Read-Only access is allowed to the file
- whether the file is to be backed-up

The attributes that are set depend on which of the R and A options are selected. If no option is selected, MS-DOS displays the attributes are are currently set.

All files are created allowing Read-Write access, ie. with the Read-Only attribute cleared. Setting this attribute allows just Read-Only access to the file until this attribute is cleared through another ATTRIB command.

The Archive attribute is set when you change a file and cleared when the file is backed up. Setting the Archive attribute with an ATTRIB command makes the file appear to have been changed and it will be backed-up by the next XCOPY command even though it has not changed. Clearing this attribute makes the latest version of the file appear to have been backed up and so it won't be backed up by the next XCOPY command even though it has been changed.

● To set attributes

Form **ATTRIB** *[+R -R] [+A -A][d:][\][path\]filename.filetype*

Options

+R	Make the file Read-Only
-R	Make the file Read-Write again
+A	Force the file to be backed-up even if it hasn't been changed
-A	Stop the file from being backed-up even though it has been changed

● To display attributes

Form **ATTRIB** *[d:][\][path\]filename.filetype*

Notes You can use a file name template, constructed using the wildcard characters * and ? (see Part I) to display the attributes of a number of files simultaneously.

Examples

- *To make the file MYFILE.TXT in the default directory Read-Only, use the command line:*

```
ATTRIB +R MYFILE.TXT
```

(assuming that the external command ATTRIB is stored either in the default directory or in a directory that MS-DOS automatically searches - see Section 7.2.4).

If MYFILE.TXT is a different directory, for example DIR1 (a subdirectory of the Root directory), the command line would be:

```
ATTRIB +R \DIR1\MYFILE.TXT
```

- *To make MYFILE.TXT Read-Write again, use the command line:*

```
ATTRIB -R MYFILE.TXT
```

if it is in the default directory or, if it is in DIR1, use:

```
ATTRIB -R \DIR1\MYFILE.TXT
```

- *To ensure that the file MYFILE.TXT in the default directory will be processed by the next XCOPY command, set the file's Archive attribute with the command line:*

```
ATTRIB +A MYFILE.TXT
```

To stop it being processed, clear the file's Archive attribute with the command line:

```
ATTRIB -A MYFILE.TXT
```

If the file is in DIR1 (a subdirectory of the Root directory), clear its Archive attribute by using the command line:

```
ATTRIB -A \DIR1\MYFILE.TXT
```

- *Display the attributes of all the files in the default directory with the command:*

```
ATTRIB *.*
```

Displayed on the screen would be something like this:

```
A>attrib *.*

R       A:\SPREAD.COM
A       A:\SPREAD.DAT
A       A:\RECORDS.OLD
A       A:\RECORDS.NEW
```

Display the attributes of all the .TXT files in \DIR1 (a subdirectory of the Root directory) with the command:

```
ATTRIB \DIR1\*.TXT
```

BACKUP

External command

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

Backup files from one disk to another

This command can back up files from one type of disk to another. For example, from hard disk to floppy disk. It will also back up one floppy disk to another, even if the disks have a different number of sides or sectors.

Form BACKUP *drive1:[path][filename] drive2:[/S][/M][/A] [/F]*
[/d:date] [/t:time] [/L:[[drive:][path]filename]]

where:
drive 1 is the disk drive that you want to back up

drive 2 is the target drive to which the files are backed up

Notes

i) If you are sharing files, MS-DOS will only let you backup files to which you have access.

ii) BACKUP will erase the old files on a backup disk, before it adds new files to it. You can prevent this by using the /a switch.

Options The BACKUP command accepts the following switches:

/s	To back up subdirectories
/m	To back up only those files which have changed since the last back up.
/a	To add files to be backed up, to those already on the backup disk. It does not erase old files on the back up disk. If you do not use the /a switch backup erases the old files on a backup disk, before adding new files to it. You cannot use this switch if files exist on the disk, which were backed up using BACKUP command from version 3.2 (or earlier) of MS-DOS.
/f	Formats the target disk, if it is not already formatted. This switch will only function while the MS-DOS FORMAT command is accessible from the current path.

/d:*date*	Will only back up files which were modified on or after date.
/t:*time*	Will only back up files which were modified on 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 which contains the files being backed up.

Backup log file

The format used by the backup log file is:

- 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.

Notes

i) If the backup log file already exists BACKUP will append the current entry to the file.

ii) 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 place where it was backed up. BACKUP displays the name of each file as it is backed up.

Example

To back up all the files in a directory called john\manual, on drive C, to a blank, formatted disk in drive A, type:

```
backup c:\john\manual a:
```

Additional Information:

The BACKUP program returns the following exit codes:

```
0    Normal completion
1    No files were found
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 command, IF, for error processing which is based on the error level returned by BACKUP.

BATCH

BATCH *[d:][\][path\]filename [value[value...]]*

Run a batch of commands

BATCH processing executes a sequence of commands stored in a specified Batch file. The commands are executed in the order they appear in the Batch file.

The commands in the Batch file can be:

- either standard MS-DOS commands
- or special BATCH subcommands.

These commands can make use of the dummy parameters %0...%9 standing in for values which are supplied when the Batch file is run.

If the Batch file is called AUTOEXEC.BAT and stored in the root directory of the Startup disk, the sequence of commands it contains will be executed immediately after the operating system has been loaded.

Form *[d:][\][path\]filename [value-2...]]*

where filename is the filename of the Batch file and replaces parameter %0 and value-1 is to replace the parameter %1, value-2 is to replace the parameter %2, and so on.

Notes The Batch file must have the filetype BAT. However, the BAT filetype is not included in the command line, that is, when you type the command BATCH to run a batch file, do not type BAT.

The Batch file can contain the names used to represent environment strings as parameters (see Section 7.1). These parameters are included in the file as %name% and they will be replaced when the Batch file is run by the string currently associated with the name.

The execution of a Batch file can be interrupted by pressing either Ctrl-Break or Ctrl-C. A message then appears on the screen asking if you want to abandon the remaining commands.

- If you type Y, the remainder of the Batch file is ignored and the system prompt appears on the screen.
- If you type N, only the current command is abandoned.

If you remove the disk holding the Batch file from its drive for the execution of one of the command lines, MS-DOS will prompt you to replace this disk before the next command line is executed.

Special BATCH subcommands

CALL

Used to call a Batch file as a subroutine of another Batch file. When the called file has been executed, the commands in the orignial Batch file continue to be executed, from the one following the CALL command.

ECHO

Turns on or off screen display of MS-DOS commands as they are executed.

FOR

Repeats the same or similar command line a number of times.

GOTO

Directs MS-DOS to a special point in the file, marked by a label.

IF

Executes a command only if a statement is true.

IFNOT

Executes a command only if a statement is false.

PAUSE

Stops further execution of a Batch file until a key on the keyboard is pressed.

REM

Displays a special message on the screen.

SHIFT

Assigns new values to Batch file parameters.

These subcommands are described in detail in the following pages.

The Batch subcommands FOR and IF can also be used outside Batch files.

Examples

- *A Batch file called RUNLOTUS.BAT that:*
  - *directs printer output to the Serial Interface on the back of your PC*
  - *makes a directory called \123FILES the current directory on the default drive*
  - *runs Lotus 1-2-3*

  would contain the lines:

```
MODE LPT1:=COM:
CD \123FILES
LOTUS
```

  Set up this simple Batch file either by using the PC text editor RPED or with a COPY command.

- *In your batch file MYBAT.BAT, you might represent the filename of a file you want to copy by %1 and the filename of the copy by %2. You might therefore have a line in the file like this:*

```
COPY %1.TXT B:%2.NEW
```

  When you run the Batch process, you follow the name of the Batch file by the values you want the parameters to represent. MS-DOS then assigns values to the parameters in order, starting with %0 which is assigned the drive, directory and filename of the Batch file. %1 takes the first value in your list, %2 the second, and so on. So running the Batch file MYBAT.BAT with the command:

```
A:MYBAT File1 File2
```

  would replace each %0 in MYBAT with `A:MYBAT`*, each %1 with* `File1` *and each %2 with* `File2`*. So MS-DOS would actually obey the command:*

```
COPY File1.TXT B:File2.NEW
```

BREAK

Internal Command

BREAK ON|OFF

Sets how often MS-DOS checks for Ctrl-Break

The way to stop the program you are running is to press either Ctrl-C or Ctrl-Break on the keyboard. MS-DOS normally checks whether Ctrl-C or Ctrl-Break has been pressed when it is taking input from the keyboard or sending output to the screen or to a printer.

The BREAK command lets you extend the number of occasions on which MS-DOS checks for Ctrl-C or Ctrl-Break to other functions, for example reading data from a disk or writing data to a disk. However, this slows down each program's speed of operation slightly and so whenever this extra checking is not needed, it should be turned off again.

- **To increase the number of occasions**

Form BREAK ON

- **To return to standard checking**

Form BREAK OFF

Example

You want to stop the next program if it goes wrong, yet you know that for much of the time it will be taking input from disk and sending all its output to disk. The command line you need is:

```
BREAK ON
```

When you are happy with the program and would prefer it to run a little bit faster again, type the command line:

```
BREAK OFF
```

CALL

Internal subcommand

Batchfile subcommand

`call` *[d:][path]batchfile*

Call a batch file from within a batchfile

Whilst running a batchfile, you can use CALL to call up another batchfile. You do not have to exit the original batchfile. This means a batchfile can be used like a command within a batchfile. When the called batchfile has been executed the commands in the original batchfile continue to be executed.

CALL can be used from any line inside a batchfile. It is limited only by available memory.

Form `call` *[d:][path]batchfile*

where: `batchfile` is the batch file you want to call

The filetype of a batch file must be `bat`, for example, CONDENSE.BAT.

Notes:

i) Do not use pipes and redirection symbols with the CALL command.

ii) A batch file can recur within itself. That is, a batch file can call itself to run again; a termination condition should be included unless you want the batch file to run more than once. If you do not include a termination condition, press Ctrl-C to terminate the command when you are ready.

Example:

To run a batch file called CLIFF.BAT from within a batch file called ROBERT.BAT, use the following command within ROBERT.BAT:

```
CALL CLIFF
```

CHCP

External command

chcp*[nnn]*

Display or change the current code page for the command processor

The CHCP command accepts one of the two prepared system code pages as a valid code page. If you select a code page which has not been prepared for the system, an error message will be displayed. If you just type the CHCP command without a code page number, CHCP displays the active code page for the system.

For detailed information on Code Page Switching, see Part one of this manual.

Form CHCP*[nnn]*

Specify the parameters:
where [nnn] is the desired code page.

If it is not specified, the operating system code page is displayed.

You may select any one of the prepared system code pages defined by the COUNTRY= command in CONFIG.SYS.

Options Valid code pages:

VALUE	CODE PAGE
437	United States
850	Multilingual
860	Portuguese
863	French-Canadian
865	Nordic

CHCP messages:

If you type CHCP the message will be similar to:

```
Active code page: 850
```

If you select a code page that is not prepared for the system, the message will be similar to:

```
code page 865 not prepared for system
```

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

```
Code page 865 not prepared for device  xx.
```

The CHCP command may need to access the COUNTRY.SYS file. If the file cannot be found, the following message is displayed:

```
File not found
```

Notes

i) Using the COUNTRY = statement in the CONFIG.SYS file or the NLSFUNC command, you can tell DOS where to find COUNTRY.SYS.

ii) The NLSFUNC command must be loaded prior to issuing the CHCP command.

Example:

To set the code page to 437 (United States), type the following command:

```
chcp 437
```

Commands *The other MS-DOS commands which support code page switching are:*

`nlsfunc`	Loads the file containing country-specific information.
`select`	Installs MS-DOS on a new floppy disk with the selected country-specific information and keyboard code.
`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 use this command to select an alternative definition file (other than the default, KEYBOARD.SYS, file) if alternatives exist.
`mode`	- 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
`graftabl`	- Displays an extended character set when using display adaptors in graphics mode.

`country`	Identifies the country in which you work or live
`device`	Installs device drivers in the system, including:

`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

CHDIR

Internal Command

`CHDIR` *[d:][\]path*

Change the current directory

CHDIR is used to:

- **change the current directory of a particular drive**
- **display the current directory path of a particular drive**

If the path you give starts from the root directory, you will need the optional backslash. If it starts from the current directory, do not include this backslash.

CHDIR can be abbreviated to CD.

- **To change the current directory**

Form `CHDIR` *[d:][\]path*

or `CD` *[d:][\]path*

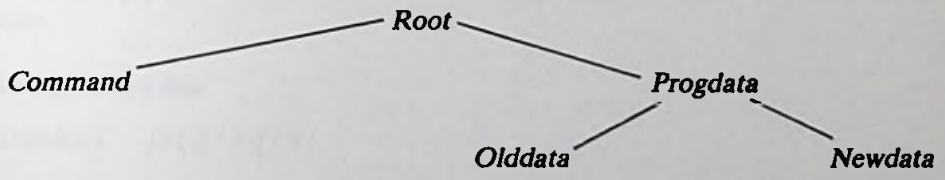
- **To display the current directory path**

Form `CHDIR` *[d:]*

or `CD` *[d:]*

Examples

Suppose this was the pattern of directories on the disk in Drive B:



- *You want the Progdata directory - a subdirectory of the Root directory on Drive B -to be the current directory on Drive B. The command line you need is:*

```
CD B:\PROGDATA
```

Note the use of the backslash to tell MS-DOS that Progdata is a subdirectory of the Root directory on this disk.

- *After using Progdata as the current directory for a while, you next want to move on to using Newdata. The command line to use is:*

  ```
  CD B:NEWDATA
  ```

  Note that there is no backslash between the colon and the name of the directory in this command line, because Newdata is a subdirectory of the current directory.

- *to reset the current directory on the default drive to the Root directory, type the command line:*

  ```
  CD \
  ```

- *You want to find out which directory on Drive C is the current directory. The command line to type is:*

  ```
  CD C:
  ```

  and the response might be:

  ```
  C:\DIR1
  ```

CHKDSK

External Command

CHKDSK *d:[\][path\][filename.filetype][/F][/V]*

Check disk for errors and file for non-contiguous storage areas

CHKDSK produces a status report on a disk and optionally on one of the files on the disk as well. This report summarises how the storage space on the disk is divided between directories, files, hidden files and free space and how parts of the specified file are allocated to different non-contiguous storage areas on the disk. (Re-using areas of the disk freed by deleting a file usually leads to the new files being stored in a number of separate areas of the disk.)

If CHKDSK encounters any places on the disk where files or directories have been corrupted, it will display appropriate error messages on the screen. Normally all these messages are displayed at the end of the process, but if you choose the /V option, these messages will be displayed as the errors are found.

CHKDSK's main role is to report errors but there are some problems that it can overcome (possibly with some loss of data) as it processes the disk. These are:

- **errors that cause parts of the directory tree to become inaccessible**
- **errors in the table that holds details of where the separate parts of a file are stored (the File Allocation Table)**

If you choose the /F option, places where the directory tree has become inaccessible will be removed from the directory tree and files will be cut short at the point the error appears in the File Allocation Table. However, sections of files that become detached from the main file are recovered and stored in files called FILE*nnnn*.CHK. If these broken files are text files (or word processor files), you can then use a text editor of a word processor to recover most of your data.

- **To report on a disk**

Form CHKDSK *[d:][/F][/V]*

- **To report on a file**

Form CHDSK *[d:][\][path\][filename.filetype][/F][/V]*

Options /F Fix errors as the disk is processed

/V Display messages as the disk is processed

NOTE: You can send the output from CHKDSK to a file rather than display it on the screen, but then you cannot use /F option

Examples

- *To check the disk in the default drive for errors, use the command line:*

```
CHKDSK
```

(assuming that the external command CHKDSK is either in the default directory or in a directory that MS-DOS automatically searches - see Section 7.2.4)

If you want to check the disk in Drive B, the command line to use will be:

```
CHKDSK B:
```

Either of these commands will produce a report something like this:

```
A>chkdsk

    730112 bytes total disk space
     53248 bytes in 3 hidden files
    625664 bytes in 56 user files
     51200 bytes available on disk

    655360 bytes total memory
    591552 bytes free
```

No attempt will be made to correct for any errors that CHKDSK finds.

- *To tell MS-DOS to check the disk in Drive B and correct as far as possible for the errors that are detected, use the command line:*

```
CHKDSK B:/F
```

Parts of the directory tree that have become inaccessible will be removed from the tree and files that have been damaged will be cut short, with the removed sections recovered and stored in files called `FILEnnnn.CHK`. *You may then be able to use a text editor or a word processor to re-unite these with the main part of the file.*

- *To discover how many separate storage areas of the default disk are used to store a file called MYFILE.TXT in the DIR1 directory, use the command line:*

```
CHKDSK\DIR1\MYFILE.TXT
```

The result you see on the screen could be something like this:

```
A>chkdsk \dir1\myfile.txt

      730112 bytes total disk space
        5124 bytes in 5 directories
      124928 bytes in 20 user files
      600060 bytes available on disk

      655360 bytes total memory
      591552 bytes free

A:\DIR1\MYFILE.TXT
Contains 2 non-contiguous blocks
```

CLS

CLS

Clear the Screen

CLS clears the screen and leaves the cursor in the top lefthand corner of the screen.

Form CLS

Note If the screen has been working in reverse video ('black' characters on a 'white' background), it will go back to its normal mode of operation ('white' characters on a 'black' background) after the screen is cleared.

Example

You may have the following display on your screen after having typed the command DIR:

```
 Volume in drive C is PC2086UK1
 Directory of  C:\

COMMAND  COM     25276 24-07-87  12:00a
ANSI     SYS      1647 24-07-87  12:00a
CONFIG   SYS        18 27-07-88   2:58p
COUNTRY  SYS     11254 24-07-87  12:00a
DRIVER   SYS      1165 24-07-87  12:00a
KEYBOARD SYS     19735 24-07-87  12:00a
PRINTER  SYS     13559 24-07-87  12:00a
RAMDRIVE SYS      6481 24-07-87  12:00a
AUTOEXEC BAT        58 27-07-88   2:58p
ASSIGN   COM      1530 24-07-87  12:00a
BACKUP   COM     29976 24-07-87  12:00a
CHKDSK   COM      9819 24-07-87  12:00a
COMP     COM      4183 24-07-87  12:00a
DEBUG    COM     15866 24-07-87  12:00a
DISKCOMP COM      5848 24-07-87  12:00a
DISKCOPY COM      6264 24-07-87  12:00a
EDLIN    COM      7495 24-07-87  12:00a
FDISK    COM     48919 24-07-87  12:00a
FORMAT   COM     11671 24-07-87  12:00a
GRAFTABL COM      6136 24-07-87  12:00a
GRAPHICS COM     13943 24-07-87  12:00a
```

If you then type CLS the screen will clear, leaving a system prompt and flashing cursor.

COMMAND

`COMMAND` *[d:][path][ctty-dev][/E:nnnn][/P][/C string]*

Start the command processor

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

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

Form `COMMAND` *[d:][path][ctty-dev][/E:nnnn] [/P][/C string]*

Where `ctty-dev` allows you to specify a different device (such as AUX) for input and output.

Specify the parameters:

The COMMAND switch `/e:` *nnnn* specifies the environment size. *nnnn* is the size in bytes, ranging from 160 to 32,768. MS-DOS rounds this number up to the next logical paragraph boundary. The default value is 160 bytes.

If *nnnn* is less than 160 bytes, MS-DOS defaults to 160 bytes and displays the following message:

```
Invalid environment size specified
```

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

`/P`	keeps the secondary command processor in memory and does not automatically return to the primary command processor.
`/C string`	Tells the command processor to perform the command, or commands, specified by string and then to return automatically to the primary command processor.

Example:

If you type:

```
command /c chkdsk b:
```

the command will tell the processor to

- *Start a new command processor under the current program*
- *Run the command CHKDSK b:*
- *Return to the first command processor*

COMP

External Command

`COMP` *[d:][\][path]filename.filetype[d:][\][path]filename.filetype*

Compare files

The COMP command is used to make a byte-by-byte comparison between two files. If the files are not expected to be exactly equal (in content AND size) to one another, the FC command should be used instead.

Form `COMP`

You will be prompted to enter the names of each of the files to be compared. If the file is on the current directory you need only enter filename.filetype.

If the file is on another disk or directory, enter the drive and pathname.

If you enter a drive letter only for the second file, the command looks on that drive for a file having the same filename and filetype as the first.

Mismatches are reported in the format:

```
Compare error at OFFSET XXXX
File1 = YY
File2 = ZZ
```

Where `XXXX` is the number of bytes into the file in hex, where the difference exists.

YY and ZZ are the hex codes for the unequal characters.

If ten unequal comparisons are detected, the command terminates with the display:

```
10 mismatches - ending compare

Compare more files  (Y/N)?
```

if the files are not EXACTLY the same size, the message:

```
Files are different sizes
Compare more files (Y/N)?
```

is displayed.

COPY

`COPY [d:][\]path\]source[/A][/B][d:][\][path\]destination[/A][/B][/V]`

Copy files

COPY is used:

- **To make copies of individual source files**
- **To make a single copy that combines a number of separate source files**

where the source file can be either a disk file or data taken from an Input Device such as the keyboard or the Serial Interface. Similarly, the destination for the copy can be either a disk file or an Output Device such as the display screen or a printer.

The COPY command can therefore be used, for example, to

- copy disk files either on the same drive or between different drives
- display a disk file on the screen or print it out on a printer
- create a file from text typed in at the keyboard

The copy of a disk file can either have the same name or a different name to the source file, although copies stored in the same directory must have a different name to the source.

A group of files can be copied with a single COPY command if you can create a file name template for the files that you want copied (see Section 6.2.2). These files should all be in the same directory.

- **To copy a disk file to another disk file**

Keeping the same file name:

Form `COPY [d:][\][path\]source[d:][\][path\]destination[/V]`

Changing the file name:

Form `COPY [d:][\][path\]source[d:][\][path\]destination[/V]`

Copying to the current directory of the default drive:

Form `COPY [d:][\][path\]source`

Eg COPY A:*.* C:\Directory*.*

Note These copies retain the date and time stamps of the original file. If you are just copying the one file, you can give the destination file a new date and time stamp by ending the source file specification with

`+,,`

in the special case of copying to the current directory on the default drive, you just need to add the +

- **To copy to an output device**

Form `COPY` *[d:][\][path\]source output-device*
where *output-device is:*

`CON`	Display screen
`PRN` or `LPT1`	A printer attached to the parallel printer port on the back of your PC
`AUX` or `COM1`	The Serial Interface
`NUL`	dummy output device used eg. when testing a program

- **To copy from an Input Device**

Form `COPY` *input-device [d:][\][path\]destination[/V]*
where *input-device is:*

`CON`	The keyboard
`AUX` or `COM1`	The Serial Interface

Notes In general, everything is recorded in the destination file until Ctrl-Z is entered. The exceptions are line-editing keys such as **[F1]** and **[←]** used when typing information at the keyboard.

If the destination file specification matches an existing file on the destination drive, the copy will be made and then the existing file will be deleted.

- **To make a file that combines data taken from a number of sources**

Form `COPY` *[/A][/B]source-1+source-2[+source-n...][path\] [destination][option[option]]*

where source-n is: *[d:][\][path\]filename.filetype[option] or input-device*
destination is: *[d:][\][path\]filename.filetype[option[option]] or output-device.*

Note You can use a file name template (see Part 1) to specify a group of disk files that are to be combined. These files all have to be stored in the same directory and they are combined in the order in which they appear in the directory.

If you leave out the destination part of this command, the combined file will replace the first source file in your list.

Options **Source**

`/A`	The file is treated as an ASCII text file. The contents are copied up to, but not including, the first Ctrl-Z (end-of-file) character. The remaining characters are not copied.
`/B`	The entire file is copied, regardless of any Ctrl-Z (end-of-file) character found in the file.

Destination

`/A`	The file is treated as an ASCII text file. A Crtl-Z (end-of-file) character is added to the end of the file.
`/B`	No Ctrl-Z (end-of-file) character is added to the file.
`/V`	The destination file is compared with the source file to verify the accuracy of the copy.

Notes The `/A` and `/B` options each apply to the file they follow and to the remaining files listed in the COPY command unless another `/A` or `/B` option is specified. The first source file can be preceded or followed by `/A` or `/B`.

The `/A` and `/B` options are only needed when ASCII and binary files are combined. They are not necessary if you are copying data, program or Batch files.

Examples

- *To copy a file called MYFILE.TXT, in the current directory on the default drive, to the current directory on Drive B and store it with the same filename and filetype, use the command line.*

```
COPY MYFILE.TXT B:
```

To store the copy under the name YOURFILE.TXT, use the command line:

```
COPY MYFILE.TXT B:YOURFILE.TXT
```

- *To copy all the files that match the file name template YOUR*.* in the default directory to the current directory on Drive B and store them with the same names but with YOUR replaced by MY, use the command line:*

  ```
  COPY YOUR*.* MY*.*
  ```

  Be sure in making such a copy, that none of the new names will be invalid, ie. that your new template does not lead to filenames more than eight (8) characters long or filetypes more than three (3) characters long.

- *To copy the file MYFILE.TXT as above but this time store the new copy with a new date and time stamp, use the command line:*

  ```
  COPY MYFILE.TXT+,, B:
  ```

- *To use the COPY command to print out the file MYFILE.TXT, use the command line:*

  ```
  COPY MYFILE.TXT PRN
  ```

- *To take input from the Serial Interface and place this in the file NEWFILE, use the command line:*

  ```
  COPY AUX NEWFILE
  ```

  All the data input through the Serial Interface will be put into this file until a Ctrl-Z character is sent. This signals the end of the file.

- *To combine the files MYFILE.1 and MYFILE.2 in a new file called NEWFILE, use the command line:*

  ```
  COPY MYFILE.1+MYFILE.2 NEWFILE
  ```

- *To append MYFILE.2 to MYFILE.1, use the command line:*

  ```
  COPY MYFILE.1+MYFILE.2
  ```

- *To append data input through the Serial Interface to MYFILE.1, use the command line:*

  ```
  COPY MYFILE.1+AUX
  ```

  All the data input through the Serial Interface will be added to the file until a Ctrl-Z character is transmitted, signalling the end of the file.

COUNTRY

COUNTRY=*xxx[,[yyy][,[d:]filename]]*

Allows MS-DOS to use international time, date, currency and case conversions

COUNTRY is a configuration command. It identifies, to MS-DOS, which country's character set you intend to use.

NOTE: It is the responsibility of the Application Progams you use to obtain this information from MS-DOS. Not all such programs will do this, therefore many Application Programs will appear not to honour the country-specific information supplied by MS-DOS.

FORM COUNTRY=*xxx[,[yyy][,[d:]filename]]*

where:

xxx	is the (telephone) country code
yyy	is the code page for the country
filename	is a file containing country information

The default value is the USA setting.

The valid country codes are as follows:

Arabic countries	785
Belgium	032
Denmark	045
English (International)	061
Finland	358
France	033
French-Canadian	002
Germany	049
Isreal	972
Italy	039
Latin America	003
Netherlands	031
Norway	047
Portugal	351
Spain	034
Sweden	046
Switzerland	041
United Kingdom	044
United States	001

Example

To change to Italian conventions of currency, time, date and case, set COUNTRY to Italy (039), insert a command line of the following form in your CONFIG.SYS file:

```
COUNTRY=039
```

Country.sys

The COUNTRY.SYS file contains country specific information.

It is the default setting for the filename parameter in the COUNTRY command. That is, if you do not specify the filename when using the COUNTRY command, MS-DOS will use the COUNTRY.SYS file for country specific information.

Note that it is highly recommended that you provide a complete path to this file, for example:

```
c:\dos\country.sys
```

CTTY

CTTY *device*

Change the device used to issue commands

Normally, all command lines are typed in at the keyboard. The CTTY command tells MS-DOS to expect, in future, all command lines to come from a different input device. For example, the PC's auxiliary input device if, for example, you want to type in command lines from a terminal attached to the Serial Interface.

- **To use an auxiliary device to issue commands**

Form `CTTY AUX`

- **To again use the keyboard for issue commands**

Form `CTTY CON`

Notes The changeover to using the other device happens immediately.

The command, which allows you to go back to using the keyboard to issue commands, has to be entered from the auxiliary device currently being used for all commands. If you get into difficulties in returning to normal operation, you can always reset your PC by holding down **[Ctrl]** and **[Alt]** and pressing **[Del]** on your PC's keyboard.

Some programs take their input directly from your PC's hardware. The CTTY command has no effect when this is the case.

DATE

Internal Command

DATE *[dd-mm-yy]*

Set or read the date

DATE is used either to display or set the current date. It is used to date stamp files.

Any date you type in is checked before it is accepted. If the date is invalid, you will be asked to type in a new date.

Form DATE

or DATE *dd-mm-yy*

where *dd* represents the day (01...31), *mm* the month (01...12) and *yy* the last two digits of the year (00...99).

Note If you type DATE, the current setting is displayed and then you are asked to type in a new date. If the date is correct, just press [↵].

Example

To set the date to be lst November 1986, either use the command line

```
DATE 01-11-86
```

which does the whole job in one go, or just:

```
DATE
```

If you use the second command, you will see something like this on the screen:

```
A>date
Current date is Sat 12-11-1986
Enter new date (dd-mm-yy):

Type:

01-11-86  [↵]
```

to set the date.

DEL
ERASE

Internal Command

DEL *[d:][\][path\]filename.filetype*
ERASE *[d:][\][path\]filename.filetype*

Delete a file

Del and ERASE are used to remove one or (by using a file name template) many files from a directory, freeing both directory space and data storage space for future files.

Once a file has been deleted, it cannot be retrieved.

Form DEL *[d:][\][path\]filename.filetype*
ERASE *[d:][\][path\]filename.filetype*

Notes If you use the file name template * . * (or equivalent) - implying that you want to erase all the files in the directory - MS-DOS asks you to confirm that you want to delete all these files by displaying the message:

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

Type Y to delete all the files; N to cancel the command.

This is the only instance in which you are asked to confirm the command before files are erased. Any other file name template causes all Read-Write files with names that match the template to be deleted immediately.

Examples

- *To delete a file called OLDDATA.1 in the default directory, type either the command line:*

```
DEL OLDDATA.1
```

or

```
ERASE OLDDATA.1
```

These command lines are exactly equivalent.

If the file you wanted to delete was in the current directory on Drive C, your command line would become:

```
DEL C:OLDDATA.1
```

- *To clear the \DIR1 directory (a subdirectory of the Root directory on the default drive) of all the files stored in it, you might use the command line:*

```
DEL \DIR1\*.*
```

Following this command, you will see the message:

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

Type Y *to delete all the files; type* N *to abandon this action.*

These commands will not necessarily clear the directory completely; any subdirectories will not be affected.

DIR

Internal Command

`DIR [d:][\][path\][filename.filetype][/P][/W]`

Display directory

The DIR command is used to:

- **display details of the files in a directory**
- **display details of a particular file, a group of files or all the files in the directory**
- **test whether a particular file or group of files is present in a particular directory**
- **list all the files in a directory that match a given wildcard file specification**

Alongside this information, you will also see:

- the disk label of the disk holding the files (if any)
- the amount of free space on the disk

Files are listed in the order they appear in the directory.

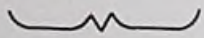
The information can be displayed in a number of different forms. The style of display used, depends upon which of the optional parameters /P and/W are selected. The version with no parameters gives a list of each file's filename and filetype together with its size in bytes and the time and date it was last changed. If this list is more than one screenful, the display will continuously scroll up until all the directory has been displayed.

- **To display details of all the files and directories in a directory**

Form `DIR [d:][\][path\][/P][/W]`

- **To display details of a single file or a group of files**

Form `DIR [d:][\][path\]filename.filetype[/P][/W]`



File specification of particular file or wildcard file specification of a group of files

- **To display details of all the files with the same filename**

Form `DIR` *[d:][\][path\]filename*`[/P][/W]`

Note Any directories with this name will also be displayed.

- **To display details of all files with the same filetype**

Form `DIR` *[d:][\][path\].filetype*`[/P][/W]`

Options

`/P`	The display pauses after every screenful of information. Press a character key to see the next screenful.
`/W`	Just the filenames and filetypes, with five files catalogued on each line.

Examples

- *To display full details - ie. filename, filetype, size and date when last changed - all the files in the default directory (ie. the current directory on the default drive), type the command line:*

```
DIR
```

To display just the names of these files, five to a line, use the command line:

```
DIR/W
```

- *To display, five to a line, the names of all the files in the current directory on Drive C, use the command line:*

```
DIR C:\W
```

- *To display full details - i.e. filename, filetype, size and date when last changed - of the files in the Root directory on Drive C, use the command line:*

```
DIR C:\/P
```

the `/P` *ensuring that the output will pause at the end of each screenful of information. Press any character key (for example, the Space bar) to see the next screenful.*

- *To display full details of just those files on the default drive that have a filename starting with the letter D and the filetype MY, use the command line:*

  ```
  DIR D*.MY
  ```

  *Note the use of the wildcard character * to create a file name template against which to match the names of all the files in the directory. * represents any valid combination of characters including, in this instance, blank.*

- *To search the \DIR1 directory for files with the filetype MY, use the command line:*

  ```
  DIR \DIR1\.MY
  ```

  *DIR interprets this command line as if it had the wildcard character * in front of the dot, allowing the template to match any valid filename.*

- *To search the \DIR1 directory for files and directories with names starting MY, use the command line:*

  ```
  DIR \DIR1\MY*
  ```

  DIR interprets this command line as if it had . at the end of the command line, allowing the template to match any valid filetype.*

DISKCOMP

External Command

`DISKCOMP` *source-drive*`:` *target-drive*`:` *[/1][/8]*

Compare contents of target drive against contents of source drive

Note that `DISKCOMP` and `DISKCOPY` apply only to floppy disks. You must use `COPY`, `XCOPY`, `BACKUP` and `RESTORE` to make copies between hard disks and floppy disks.

The DISKCOMP command is used to make a track-by-track comparison between two disks. These disks can either be in different drives or in the same drive. DISKCOMP first checks that the disks have the same format and then checks each track on the target disk against the corresponding track of the source disk. Where it finds a difference, it displays a `Compare error` message giving the track number and the side where the difference was found.

DISKCOMP normally compares both sides of the disks and all nine sectors on each track, but you can limit the comparison to just one side of the disks or to just the first eight sectors on each track by selecting the appropriate options. It compares a section of the disks at a time, reading first the source disk and then checking the target disk against this.

After the comparison has been completed, DISKCOMP asks you if you want to compare another pair of disks. Type **[Y]** [↵] to compare another; type **[N]** [↵] to leave DISKCOMP.

Form `DISKCOMP` *source-drive*`:` *[target-drive:][/1][/8]*

Options

/1	Compare just the first side of the disks
/8	Compare just the first eight sectors of each track

Notes If you do not specify a target drive, this is taken to be the default drive.

If the target drive is the same as the source drive, DISKCOMP will prompt you when to insert the source disk and when to insert the tarket disk as it compares the disks section by section.

You cannot use DISKCOMP on a drive that has been ASSIGNed, JOINed or SUBSTituted (see Section 7.1).

Some disk drives will not support the /1 and /8 options. See the table in Part 1

If your default drive is A>, then when DISKCOMP is terminated it will display the message:

```
Insert disk with /COMMAND.COM in DRIVE A
and strike any key when ready
```

At this prompt you should insert either your SYSTEM disk or any other disk which contains a copy of the COMMAND.COM file and press the [↵] key. MS-DOS will reload the internal commands and then display the A>. (No message will appear if COMMAND.COM is on disk already in drive A; it will be reloaded automatically).

Examples

- *To compare the disk in Drive A with the disk in Drive B, use the command line:*

```
DISKCOMP A: B:
```

(assuming that the external command DISKCOMP is stored either in the default directory or in a directory that MS-DOS searches automatically - see Section 7.2.4)

This is the command line to use if Drive A or Drive C is the default drive, but if Drive B is the default drive, your command line can be:

```
DISKCOMP A:
```

the result of this comparison could be something like this:

```
A>diskcomp a: b:
Insert FIRST  diskette in drive a:
Insert SECOND diskette in drive B:
Press any key when ready...

Comparing 80 tracks
9 sectors per track, 2 side(s)

Compare OK
        or    Compare error on side 0, track 0

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

DISKCOPY

External Command

`DISKCOPY` *source-drive*`:` *target-drive*`:` *[/1]*

Copy contents of source drive to target drive

Note that `DISKCOMP` and `DISKCOPY` apply only to floppy disks. You must use `COPY`, `XCOPY`, `BACKUP` and `RESTORE` to make copies between hard disks and floppy disks.

DISKCOPY is used to produce a track-by-track copy of one disk on another. If the target disk is unformatted, it will format it identically to the source disk as part of making the copy.

The target drive does not have to be a different drive from the source drive. If these drives are the same, then DISKCOPY will read the source disk and then prompt you to change the disk in the drive to the target disk.

The [/1] option allows you to copy only one side of the disk.

When the copy has been completed, DISKCOPY asks you whether you want to copy another disk. Type **[Y] [** ↵ **]** to copy another; type **[N] [** ↵ **]** to leave DISKCOPY.

Form `DISKCOPY` *[source-drive:] [target-drive:] [/1]*

Notes If you only specify one drive, then the target drive is taken to be the default drive. If you do not specify any drives, DISKCOPY performs a single-drive copy that uses the default drive as both the source drive and the target drive.

If you want to use the default drive as the source drive but another drive as the target drive, you have to give the drive letters of both drives.

The target disk will be formatted before the copy is made if it has not been formatted already or if it does not have the same format as the source disk.

Some disk drives will not support the /1 option. See the table in Part 1.

If your default drive is A>, then when DISKCOMP is terminated it will display the message:

```
Insert disk with /COMMAND.COM in DRIVE A
and strike any key when ready
```

At this prompt you should insert either your SYSTEM disk or any other disk which contains a copy of the COMMAND.COM file and press the **[** ↵ **]** key. MS-DOS will reload the internal commands and then display the A>. (No message will appear if COMMAND.COM is on disk already in drive A; it will be reloaded automatically).

Example

To make a copy of the disk in Drive A, use the command line:

```
DISKCOPY A: B:
```

(assuming that the external command DISKCOPY is stored either in the default directory or in a directory that MS-DOS automatically searches - see Section 7.2.1)

This is the command line to use if Drive A or Drive C is the default drive. If Drive B is the default drive, you can use the command line:

```
DISKCOPY A:
```

If you have a single-drive PC, you will be prompted when to insert the disk for Drive A (the disk you want to copy) and when to insert the disk for Drive B (the target disk, ie. the disk you want to store the copy on). If you have a two-drive system, put the target disk in Drive B (the righthand disk drive).

If the target disk has not been formatted or has a different format to the source disk, you will see the following message on the screen:

```
Formatting while copying
```

DISPLAY.SYS

Device driver

`Device=`*[d:][path]*`display.sys con[:]=[type[,hwcp][,n,m]]`

DISPLAY.SYS is a device driver that supports code page switching for the console device.

For information about code page switching, see Part One.

Installation

To install DISPLAY.SYS, insert a command line of the following form in your CONFIG.SYS file:

Form `device=`*[d:][path]*`display.sys con[:]=[type[,hwcp][,n,m]]`

The following list of options shows how each variable is used:

OPTION	FUNCTION
`type`	The display adaptor in use. Valid values include MONO, CGA, 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.

ECHO

ECHO *[ON | OFF | remark]*

Control echoing of commands on the screen

Normally, as a Batch file is being obeyed, each command line that MS-DOS executes is displayed on the screen just before it is executed.

The ECHO subcommand lets you:

- turn the display of command lines off
- turn the display of command lines on
- display a message, regardless of whether command lines are being displayed

● **Turn the display off:**

Form ECHO OFF

● **Turn the display back on:**

Form ECHO ON

● **Display a message**

Form ECHO *[ON | OFF]remark*

Notes:

1) ECHO OFF stops REM messages and PAUSE special messages from being displayed on the screen. The PAUSE subcommand message `Press any key when ready ...` is still displayed.

2) ECHO remark displays the given message regardless of whether command line echoing is on or off. The message may be no more than 40 characters long (including spaces).

3) When ECHO is ON you can suppress the display of any line in your Batch file by putting an @ character before it.

- *You do not want any of the command lines and messages from your Batch file to appear on the screen. The first lines in your Batch file should therefore be:*

```
ECHO OFF
CLS
```

the CLS command clearing the screen ready for displaying the output from your programs.

- *You want to turn ECHO off and change the default directory, without the commands being displayed on the monitor. The command lines:*

```
@ECHO OFF
CD\MYDIR
```

in your batch file, will execute the two commands without displaying them; the first because of the @ and the second because ECHO has now been turned OFF.

The only messages that you will see while ECHO is OFF will be messages contained within special ECHO commands and the `Press any key when ready...` *that goes with a PAUSE command.*

To turn the display of messages back on, use the command line:

```
ECHO ON
```

Include this in the Batch file immediately before the first command or message that you want to see on the screen.

To display messages only at strategic stages in the Batch process, begin the Batch file as above with the lines:

```
ECHO OFF
CLS
```

At the points in the Batch file you want messages displayed, put lines of the form:

```
ECHO message
```

For example, to tell you that Stage I has been completed and Stage II is about to start, you might insert the line:

```
ECHO Stage 1 finished; Stage 2 begun
```

EDLIN

`EDLIN` *[d:][\][path\]filename.filetype[/B]*

Edit text

EDLIN is MS-DOS's text editor and can be used to modify files of ASCII characters (ie. text files). whether these are data files or BASIC programs. It can also be used to create new text files for storage on disk.

It can be used as an alternative to the PC text editor, RPED, see Chapter 6.

EDLIN opens an area of memory to act as a workspace and looks for the file you specify in the command line. If it finds the file, EDLIN copies the file into its workspace ready for editing - filling up to three-quarters of the workspace area. Any remainder can only be read into the workspace after the previous section has been edited and stored as the new version. If all the text is copied into the workspace, the message `End of input file` will appear on the screen. If EDLIN does not find the file, it displays the message `New file`. In either case, it will display a star on a fresh line when it is ready to receive editing commands. This star is the EDLIN prompt.

While you are in the editor, the commands you use are the special EDLIN ones: you cannot use any MS-DOS commands until you leave the editor. However, you can use the Command Line Editing keys (see Section 7.2) to help you prepare new lines based on existing lines of text.

The EDLIN commands copy text into and out of the workspace, edit particular lines and insert or delete groups of lines. There are also commands to move a group of lines from one place to another, to duplicate lines and to search and replace strings of characters. The commands are listed below. The lines you edit are displayed preceded by a line number which tells you how many lines down the file you are working. These numbers are kept constantly up to date and are not saved with the file. The current line you are working on is indicated by a star and line numbers may be given relative to this line (see 'Defining lines' below).

A number of EDLIN commands can be typed in one after another before the [↵] key is pressed to send the instruction for processing. The first command in a command line can be a line number, changing which line is the current line: the remainder have to work with whatever current line has been set. If the combined instruction starts with a line number, the individual commands must be separated by semicolons. If the instruction does not start with a line number, the commands can be typed in without semicolons (or any other separators) between them.

When you want to finish the edit and save the edited version, type **E [←]** after a * prompt. However, if there is not enough room for both the new version and the old version of the file (the back-up) on the disk, the new version will either be completely lost or only part of it will be saved. It is therefore advisable to check that there is enough room for both the old and new versions of the file on the disk (see Section 7.2.5).

The modified file is given the **filename** and **filetype** of the original file. However, the original file is not erased. Instead, it is renamed **filename.BAK** (replacing any file that already exists with that name). Should you want to use the old version again, you will need to give it some other filetype as many commands (including EDLIN) do not work on **.BAK** files.

Form **EDLIN** *[d:][\][path\]filename.filetype[/B]*

Notes The optional **/B** is used to tell EDLIN not to interpret Ctrl-Z characters as ends-of-files.

EDLIN commands

n

Edit Line *n* (if omitted, it is taken to be the next line). Line *n* becomes the current line and is displayed, followed by a fresh line with the same line number. The Command Line Editing keys can then be used to construct a new line based on the present version of this line.

If you do not want to change the line, just press **[←┘]** -pressing **[←┘]** when you have copied some of the old line into the new one will delete everything to the right of the current cursor position.

*[n]***A**

Append *n* of the lines of text that have so far not been copied into the workspace to the end of the text currently in the workspace. If *n* is omitted, as many lines as possible will be copied but once again, the workspace is not allowed to become more than three quarters full.

*[n],[m],/[,c]***C**

Copy a range of lines to a different part of the file.

n specifies the first line to be copied (if omitted, it is taken to be the current line).

m specifies the last line to be copied (if omitted, it is taken to be the same as *n*).

/specifies the line before which the copy is to be inserted (which can not be between *n* and *m*)

c specifies the number of copies to be inserted at this point (if omitted, one copy is made).

*[n][,m]*D	Delete a range of lines. *n* specifies the first line to be deleted (if omitted, it is taken to be the current line). *m* specifies the last line to be deleted (if omitted, it is taken to be the same as *n*).
E	End the edit, save the new file and return to MS-DOS.
*[n]*I	Insert the following lines immediately before Line *n*. If *n* is omitted, the lines are inserted before the current line; if *n* is #, the new text will be inserted after the last line currently in the workspace. Type Ctrl-C after you have typed in the last new line. To insert control codes in the file, type Ctrl-V followed by *character*, where the code you want is Ctrl-character.
*[n][,m]*L	List a range of lines. *n* specifies the first line to be displayed on the screen (if omitted, it is taken to be 11 lines before the current line). *m* specifies the last line to be displayed (if omitted, 23 lines are displayed).
*[n],m,/*M	Move a range of lines to a different part of the file. *n* specifies the first line of the range (if omitted, it is taken to be the current line). *m* specifies the last line of the range. The lines are inserted immediately before Line/.
*[n][,m]*P	Display a range of lines in pagefuls of 23 lines at a time. The last line becomes the new current line. *n* specifies the first line to be displayed (if omitted, it is taken to be the current line). *m* specifies the last line to be displayed (if omitted, 23 lines are displayed).
Q	Quit the edit, abandoning any changes made so far, and return to MS-DOS. EDLIN asks you to confirm that you do not want to save the edited version. Type Y to abandon the edit. Pressing any other key keeps you in EDLIN and allows you, for example, to End the edit and thus save the edited version.

*[n][,m][?]*R*old-string*[Ctrl-Z*[new-string]]*

Replace all occurrences of *old-string* in a range of lines with *new-string*. If both Ctrl-Z and *new-string* is omitted, EDLIN will delete *old-string* (ie.it will not replace it with anything). If both Ctrl-Z and *new-string* are omitted, *old-string* will be replaced by the last *new-string* you used.

n specifies the first line of the range (if omitted, it is taken to be the line following the current line).

m specifies the last line of the range (if omitted, it is taken to be the last line in the workspace).

Including ? makes EDLIN ask for confirmation before replacing the current instance of *old-string* with *new-string*. Press [Y] or [←┘] to confirm the change.

*[n][,m][?]*S*string* [←]

Search for the first occurrence of string in a range of lines.

n specifies the first line to be searched (if omitted, it is taken to be the line following the current line).

m specifies the last line to be searched (if omitted, it is taken to be the last line in the workspace).

Including ? makes EDLIN ask if it has found the instance of string you wanted (the whole line is displayed on the screen). Press [Y] or [←┘] to stop the search, press any other key to continue the search.

Note This must be the last command in the current instruction.

*[l]*T*[d:][\]path\]filename.filetype*

Transfer the whole of the named file into the workspace, inserting it immediately before Line 1. If 1 is omitted, it is inserted before the current line.

*[n]*W

Write the first n lines in the workspace to the disk. If n is omitted, enough lines are copied to make the workspace only one quarter full.

Defining lines There are a number of ways in which particular lines can be specified in these commands.

line-number	the actual line number. **Note:** Give any number that is greater than the last line number but less than 65534 to get the line immediately following the last line.
+number-of-lines	the number of lines the chosen line is after the current line.
−number-of-lines	the number of lines the chosen line is before the current line.
.	the current line
#	the last line

Example

You can make the running of any program particularly convenient by including the instruction to run the program in the AUTOEXEC.BAT file of your Startup disk. The necessary changes can be made to the AUTOEXEC.BAT file using EDLIN.

For example, suppose you have a two-drive system and you want to run BASIC simply by placing the WINDOWS disk in Drive A and your BASIC disk in Drive B. This involves adding the command PATH B: to the AUTOEXEC.BAT file and changing the final line to WINDOWS BASIC.APP

The steps are as follows:

1) Start editing the AUTOEXEC.BAT file with the command line:

```
EDLIN B:AUTOEXEC.BAT
```

When this file is ready for editing, you will see on the screen:

```
End of input file
```

Note: *For many purposes, where working with small files, such as batch files, the small-file editor RPED will be found more convenient to use (see Chapter 6).*

2) Press the [L] key. This displays the whole of this short file on your screen. You will see something very like:

```
1:  copy command.com c:
2:  set comspec=c:\command.com
3:  copy windows.bat c:
4:  c:window
```

3) *The PATH instruction has to be inserted before line 4, so type:*

`4I [↵]`	*- to start insert the new line*
`path b:\basic[↵]`	*- this is the text of the new line*
Ctrl-Z	*- to end the insert*

4) *Press the [L] key to see the effect of these actions. You should see something very like:*

```
1: copy command.com c:
2: set comspec=c:\command.com
3: copy windows.bat c:
4: path b:\basic
5: c:windows
```

5) *The next stage is to modify the last line, so type or press:*

`5`	*- to start editing line 5 (the current line number of the last line)*
`F3`	*- to copy the whole of the old line into the new line**
Space bar	*- to put a space after window*
`basic2.app`	*- to finish the new line*

** If WINDOWS is followed by %1 %2 %3, press [F2] instead of [F3] and then type %*

6) *Press the [L] key to see the effect of all these actions. You should see something very like:*

```
1: copy command.comc:
2: set comspec=c:\command.com
3: copy windows.bat c:
4: path b:\basic
5: c:windows basic.app
```

7) *Press the [E] key to finish the edit.*

FASTOPEN

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

Reduce time taken to open frequently-used files and directories

Form `fastopen d:[=nnn]...`

This command reduces the time it takes to open frequently-used files and directories.

Normally, opening and closing files in a complex directory structure, slows your computer down. Particularly when you run applications that use several files, such as a database application.

FASTOPEN works by tracking the location of files and directories on a disk for fast access. The name and location of all the files that are opened are recorded, then if one of these files or directories is reopened, access time is reduced.

FASTOPEN only works on hard disks and will not work over a network. It can be used with up to four hard disks at one time. For each hard disk, FASTOPEN will track a range of files or directories, from 10 to 999. The default number is 34.

Therefore, you can use the command FASTOPEN once, to specify the number of files per drive for more than one drive.

Notes:

i) You can invoke the FASTOPEN command only once.

ii) If you want to change the FASTOPEN settings restart MS-DOS.

iii) FASTOPEN needs approximately 35 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, type:

```
fastopen c:=100
```

If you want MS-DOS to track the location of up to 100 files on drive C, and up to 50 files on drive D, type:

```
fastopen c:=100 d:=50
```

FC

External command

FC *[option[option . . .]][d:][path]filename.filetype[d:][path]filename.filetype*

Compare files

The FC command is used to compare two files. The files to be compared are either treated as text files or as binary files:

- text files are compared line-by-line
- binary files are compared byte-by-byte

Note that it is not possible to use filename templates with two sets of files.

When a difference is found between two text files, FC continues to look for lines from the two text files that match. Unless instructed otherwise, tab characters are converted to spaces up to the start of the next 8-character column before the comparison is made.

When FC finds two consecutive lines in one file that match two consecutive lines in the other, it resynchronises the comparison and carries on to the next place in the files where it finds a difference.

No attempt is made to resynchronise files compared byte-by-byte.

Notes

i) There is a limit to how much FC can read when looking for lines which match. These lines are stored in the FC command's buffer. If the buffer becomes full before a match is found, FC declares the files "too different" and ends the comparison.

ii) You can control the size of the buffer.

Options

Select options to control the type of comparison. If no options are specified:

- files with the filetype EXE, COM, SYS, OBJ, LIB and BIN are treated as binary files; everything else is treated as a text file.
- the size of the buffer used for the comparison is set to 100 lines.
- tabs, spaces and upper and lower case in a text file are all significant
- two lines of the text have to be identical for the comparison to resynchronise

● **Compare files line by line**

Form `FC` *[option[option...]][d:][path]filename.filetype* *[d:][path]filename.filetype*

file to be compared *standard for comparison*

Options

`/A`	To indicate where the difference is. Display the last line previous matching section and the first line of the following matching section, but not all the lines in between. (The lines in between are represented by...)
`/C`	Treat all letters as upper case in the comparison
`/L`	Force a line by line comparison. Use when the filetype of either file is EXE, COM, SYS, OBJ, LIB or BIN.
`/LBn`	Set the buffer size to n number of lines. The default is 100 lines. The maximum possible number depends on the memory space in your PC.
`/N`	Add line numbers when reporting where the difference are.
`/T`	Don't expand tab characters to spaces. (The default is to interpret each tab character as spaces up to the start of the next 8-character column).
`/W`	Interpret all tabs and multiple spaces as single spaces; ignore spaces at the beginning and the end of any line.
`/n`	Set the number of identical lines required, before the comparison resynchronises, to n. (The default number is 2 lines).

Notes i) If FC runs out of buffer space while reading lines which do not match, FC will not be able to resynchronise the comparison. If this happens the following message will be displayed:

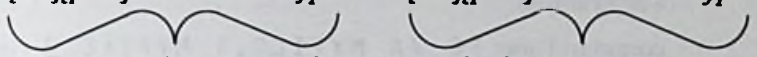
```
Resynch failed.  Files are too different
```

The comparison will then be abandoned. Use the /LBn option to increase the size of the buffer.

ii) You cannot use the `/B` option of a byte by byte comparison in a line by line comparison. If you try this MS-DOS will display the message: `Incompatible switches.`

● **Compare files byte-by-byte**

Form `FC [/B] [d:][path]filename.filetype [d:][path]filename.filetype`



Notes i) The /B option should be used to force a byte-by-byte comparison if either of the files does not have the filetype EXE, COM, SYS, OBJ, LIB or BIN.

ii) No attempt is made to resynchronise the comparison after a difference has been spotted.

Examples

To compare the two text files, MYFILE.1 and MYFILE.2, line by line, you might use the following command line:

```
FC MYFILE.1 MYFILE.2
```

This command line would search each line letter for letter, space for space with any tab characters expanded to spaces, such that the next character began the next 8 character column. If MYFILE.2 was identical to MYFILE.1 except for four additional lines, you might see a report like this:

```
A>fc myfile.1 myfile.2
****** myfile.1
When the hurly-burly's done
When the battle's lost and won
That will be ere set of sun
And where the place
Upon the heath
***** myfile.2
When the hurly-burly's done
Upon the heath
*****
```

If there are many differences between the files, the above report would be very long. In that case you should use the /A option which will give only the last line of the previous matching section and the first line of the following matching section. For example:

command line: `FC /A MYFILE.1 MYFILE.2`

response:

```
A>fc myfile.1 myfile.2/a
***** myfile.1
When the hurly-burly's done

...
Upon the heath
***** myfile.2
When the hurly-burly's done
Upon the heath
*****
```

If you want a comparison to also tell you whereabouts in the files the difference was found, use the /N option, for example:

```
FC /N MYFILE.1 MYFILE.2
```

If you want to compare the files COMMAND.EXE and COMMAND.SYS use the command line:

```
FC COMMAND.EXE COMMAND.SYS
```

FC would carry out a byte-by-byte comparison on these two files (because of their filetypes) and if any difference is spotted, no attempt will be made to resynchronise the files. So if COMMAND.SYS is identical to COMMAND.EXE except for two extra bytes, you would see a report like the following:

```
A>fc command.exe command.sys

00000646 6B 7A
00000647 1A 3F
00000648 1B 6B
00000649 23 1A
0000064A FB 1B
0000064B 00 23

fc command.sys longer than command.exe
```

FDISK

FDISK *[d:]*

Configure a hard disk

The FDISK command configures a hard disk for use with MS-DOS. A series of menus are displayed to help you create and delete partitions.

FDISK does not work on drives used in the SUBST or JOIN commands.

WARNING: Reconfiguring your disk with FDISK destroys all existing files. Make sure you have back up copies of all files on your disk before you create an MS-DOS partition with FDISK.

FORM FDISK

You can use it to:

- Check whether a hard disk is configured or formatted
- Create a primary or single DOS partition
- Format your hard disk to start MS-DOS
- Create an extended DOS partition
- Create logical drives in extended partitions
- Change the active partition
- Delete a primary DOS partition
- Delete an extended DOS partition
- Delete a logical drive in an extended DOS partition
- Display partition data
- Select the next hard disk drive for partitioning on a system with multiple fixed disks.

● **To check whether a hard disk is configured and formatted:**

1. Try to start MS-DOS from your hard disk. If it starts your hard disk is both configured and formatted and the MS-DOS system files are on the disk.
2. If MS-DOS does not start, your disk is not formatted to start MS-DOS, but it may have been configured. Check to see if the disk has been configured with FDISK:
   - Place the MS-DOS master floppy disk in drive A
   - Hold down the **[Ctrl]** and **[Alt]** keys and press **[Del]** to start MS-DOS
   - Run FDISK and select the `Display Partition Data` option to see if any MS-DOS Partitions exist

   If any partitions do exist your disk has been configured.

   If no partitions exist follow the instructions below to configure the disk.

IMPORTANT once your hard disk is configured you must format the disk with the MS-DOS FORMAT /S command before you copy files onto the disk. Otherwise you will never be able to load MS-DOS from the disk, as is the usual way.

● **How to start FDISK**

- Place the MS-DOS disk in drive A
- Type FDISK at the A>, and press the [↵] key

FDISK will display its main menu on your screen. If your computer has only one hard disk this menu will list four choices. If your computer has more than one hard disk it will list five choices:

```
***
Disk options
Current Fixed Disk Drive:1
Choose one of the following:

     1. Create DOS Partition
     2. Change active Partition
     3. Delete DOS Partition
     4. Display Partition Data
     5. Select Next Fixed Disk Drive
Enter choice:        [1]
Press Esc to return to DOS
***
```

Use the **[Esc]** key to return to the previous menu or DOS, from any of the FDISK menus.

If you wish to choose the default value on any menu press the [↵] key. To choose another value, type the value you want and press the [↵] key.

● **How to create a DOS Partition**

You must create a Primary DOS Partition before you can create an extended DOS Partition.

From the main menu select the **Create DOS Partition** option.

If your hard disk is not completely partitioned the screen will display:

```
***
Current Fixed Disk Drive: 1

1. Create Primary DOS Partition
2. Create Extended DOS Partition
3. Create Logical DOS Drives 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

Press the [↵] key to accept the default selection (1) of the above menu.

1. Creating a primary DOS Partition

Type f d i s k at the A>. The main menu will be displayed.

Select option 1 **Create DOS Partition**

Then select option 1 **Create Primary DOS Partition**

The **Create Primary DOS Partition menu** will appear:

```
***
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
***
```

Accept the default selection (Y) , by pressing the [↵] key, if you want to create a single partition, that is, if you want to use the entire hard disk for MS-DOS (up to 32 megabytes).

FDISK then displays the following message:

```
System will now restart

Insert DOS diskette in drive A:
Press any key when ready...
```

Leave your MS-DOS disk in drive A and press any key to restart MS-DOS.

If you decided to create a single partition, you should now go ahead and format your disk, using the FORMAT command. Be sure to use the same version of MS-DOS for both FDISK and FORMAT.

- **Creating more than one MS-DOS Partition**

Select option 1 **Create DOS Partition.**

Then, from the Create DOS Partition menu, select option 1, **Create Primary DOS Partition.**

You will be asked: `Do you wish to use the maximum size for a DOS partition and make the DOS partition active (Y/N)...? [Y]`

Answer N to this question.

FDISK will display a second create primary dos partition menu which asks you to specify the size of the primary dos partition:

```
create primary dos partition

current fixed disk drive:1

partition status type start end size

total disk space is 614 cylinders
maximum space available for partition is 614 cylinders

Enter partition size....[614]

Press Esc to return to Fdisk options
```

Enter the size you want your Primary Partition to be and press the [↵] key.

The space on your disk is measured in cylinders (known also as tracks). This menu shows the total number of cylinders available for a hard disk partition.

Once you have defined the size of the Primary DOS Partition, a confirmation screen will appear:

```
create primary DOS partition

current fixed disk drive: 1

partition status type  start  end  size
C:1              PRI DOS   0    xxx  xxx

primary DOS partition created

press ESC to return to fdisk options
```

The message: `WARNING! No partitions marked active` also appears on this screen because the partition you have created is not automatically made active. You will have to do this using the main menu, option 2. You will not be able to startup MS-DOS from this partition until it is made active.

- **Creating an extended DOS Partition**

Once you have created a primary DOS Partition, you should create an extended DOS Partition on your hard disk if your hard disk is larger than 32 megabytes or you want to designate one or more logical drives for the disk. Use the create DOS partition menu, option 2. and press the [↵] key. FDISK will then display the following menu:

```
create Extended DOS partition

current fixed disk drive:1

partition  status  type  start  end  size
C: 1       A      PRI DOS  0    xxx  yyy

total disk space is 1263 cylinders
maximum space available for partition
is 663 cylinders

enter partition size.....[663]

press esc to return to Fdisk options
```

This menu shows the total number of cylinders available for an extended partition. The default for the partition size is the maximum available space on the hard disk. Press the [↵] key if you want the default value. If not, type in the size (in cylinders) that you want for the partition, and press the [↵] key.

NOTE: if FDISK finds any bad tracks it adjusts the partition bounderies to avoid those bad tracks.

Creating Logical Drives in the Extended DOS Partition

Once you have created an extended DOS partition, you must specify one or more drive letters for that area of the disk.

FDISK will automatically display the create logical drives in the extended DOS partition menu after you have created an extended partition.

Choose option 3 from the Create DOS Partition menu, and a screen similar to the following will appear:

```
Create Logical DOS Drive(s)

Drv     Start      End      Size
D:      mmm        nnnn     xxx

Total partition space is 1000 cylinders.
Maximum space available for logical drive
is 600 cylinders.

Enter logical drive size ..... [xxx]

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, you may want to keep two applications in separate drives with their respective data files.

You must assign the whole of the extended partition to a logical disk drive because you cannot use an MS-DOS extended partition without a drive letter. FDISK will continue to prompt you for logical disk drive information until the whole partition has been assigned.

When you have finished this, FDISK displays this message:

All available space in the Extended DOS partition is assigned to logical drives

Press **[ESC]** to return to the main FDISK menu. From there, you can restart MS-DOS, or select another option.

Changing the Active Partition

Select option 2 of the main menu and FDISK will display information about all the partitions on your hard disk: size, position, type, and whether active or not.

There is only one active partition on a hard disk, and that is the one the computer uses when it is first turned on. All the other partitions are inactive. If you have only one partition, then it has to be active.

If you have more than one partition the screen would display information similar to the following:

```
Change active partition

Current fixed disk drive: 1

partition      status     type       start      end      size
C:   1                    PRI DOS    0           413      414
     2                    EXT DOS    414         415      200

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
```

Type the number of the partition you want to make active, and press the [↵] key. The computer will choose the currently active partition as the default, to reduce the risk of accidents.

If you had used other operating systems, such as XENIX for example, extra partitions with the type "non-dos" would be displayed. These partitions cannot be made active by FDISK. (apart from EXT.DOS which is not bootable).

If you have just one partition covering the entire hard disk, then FDISK will inform you:

```
Partition 1 is already active

Press ESC to return to FDISK options
```

In that case, you will have to press **[Esc]** to return to the main menu. No other key will have any effect.

- **Deleting the DOS Partition**

Choose option 3 from the main menu Delete the DOS Partition and FDISK will display the following menu, asking whether the partition you want to delete is a primary or extended DOS partition:

```
Delete DOS Partition

Current fixed disk drive

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
```

Enter your choice and press the [↵] key. The next menu will show the status of the partition you want to delete.

NOTE: FDISK deletes in a specific order, namely:

1 - logical drives
2 - extended partitions
3 - primary partition

If you try to delete in another order, FDISK will tell you that the deletion cannot be done while a lower element is in existence.

FDISK deletes the partition bounderies and any data that existed in that partition. Once deleted, you cannot recover the data that was on it.

NOTE: You cannot use FDISK to delete a non-DOS partition. You should delete them using the program that was used to place them on the disk.

- **Deleting Logical DOS Drives**

Choose the Delete menu option 3.

FDISK will display a screen showing information on each logical drive on your hard disk. It will ask you which one you want to delete. After making your selection, you will be asked to verify your selection.

BEWARE: If you answer Y to that question, you will not be able to recover the logical DOS drive, nor all the data it contained.

The screen will look something like this:

```
Delete Logical DOS Drive

DRV        Start      End        Size
D:         414        513        100
E:         514        613        100

Total partition size is 200 cylinders

Warning!   Data in the logical DOS drive
will be lost.  What drive do you wish
to delete.............?  [D]

Are you sure..........?  [N]
```

Press **[Esc]** to return to FDISK options

If you do not want to delete your logical DOS drive, press either the [↵] key or **[Esc]** to return to the main menu.

Otherwise, type Y and press the [↵] key. The logical drive will be deleted, and if there are more logical drives, as in the above example, FDISK will repeat the procedure.

When there are no more logical drives, you will be shown a confirmation screen and then returned to the main menu.

- **Deleting Extended DOS Partition**

Choose menu option 2 if there are no logical drives. FDISK will display a screen showing information on each partition on your hard disk. You will be asked whether or not you want to delete the Extended DOS Partition.

The screen will look something like this:

```
Delete Extended DOS Partition

Current Fixed Disk Drive: 1

Partition      Status         Type       Start      End   Size
C: 1           A              PRI DOS    0          413   414
   2                          EXT DOS    414        613   200

Warning!  data in the Extended DOS
partition will be lost.  Do you wish
to continue ..................?  [N]

Press ESC to return to FDISK Options
```

If you do not want to delete your extended DOS partition, press the [↵] key, to accept the default, or the **[Esc]** key to return to the main menu.

Otherwise type Y and press the [↵] key. The partition will be deleted, you will be shown the confirmation screen and then returned to the main menu.

- **Deleting the Primary DOS Partition**

Choose menu option 1 (but only if there is no extended partition and no logical drives) from the Delete DOS Partition menu.

FDISK will display a screen showing information on each partition on your hard disk. It will ask you whether or not you want to delete the Primary DOS Partition.

BEWARE: If you answer Y to that question, you will not be able to recover the primary DOS partition nor all the data that it contained. FDISK therefore selects N as the default.

The screen will look something like this:

```
Delete Primary DOS Partition

Current Fixed Disk Drive: 1

Partition      Status     Type       Start      End        Size
C: 1           A          PRI DOS    0          413        414

Warning!  Data in the Primary DOS
Partition will be lost.  Do you wish
to continue.................?  [N]

Press Esc to return to Fdisk Options
```

If you do not want to delete your DOS partition, press either the [↵] key to accept the default answer, or **[Esc]** to return to the main menu.

If you do want to delete your DOS partition, type Y and press the [↵] key. The partition will be deleted, you will be shown the confirmation screen and then returned to the main menu.

- **Displaying Partition Data**

Choose the fourth main menu option.

FDISK will display a screen showing information about all the partitions on the disk. It will look something like this.

```
Display Partition Information

Current Fixed Disk Drive:  1

Partition      Status     Type       Start      End   Size
C:  1          A          PRI DOS    0          413   414
    2                     EXT DOS    414        613   200

Total disk space is 614 cylinders

{ The Extended DOS size partition contains
  logical DOS drives.  Do you want to display
  logical drive information?          [Y]

Press  Esc to return to Fdisk Options
```

Note that the extra message, shown in brackets, in the above example, will only be displayed when there are logical drives present. If you press Y and the [↵] key the screen will look like this.

```
Display Partition Information

DRV  Start      End   Size
D:   414        513   100
E:   514        613   100

Press ESC to return to Fdisk Options
```

Pressing **[Esc]** will return you to the main menu.

- **Selecting the Next Fixed Disk Drive**

This main menu option only appears if you have more than one hard disk attached to your computer.

If you choose this option FDISK changes the current disk drive to the next drive. For Example, if the current disk drive is drive C, choose option 5 to change it to drive D.

You can then choose any of the FDISK options to prepare the second fixed disk for MS-DOS. Or, you could select option 5 once again to select the next drive. For example, if there is not a third fixed disk, FDISK changes the current drive from D back to C.

After you have selected the next drive, FDISK displays the main menu again. Note that there is a line near the top of the screen which looks like this:

```
Current fixed disk drive:  2
```

The activity you select will be performed on the disk shown on this line.

FIND

External Command and External Filter

`FIND` *[/V][/C][/N]* `"string"` *[d:][\][path\]filename.filetype*

Look for a string of characters in a file

FIND is used to search for a given string of characters in one or more files. These can be either files that are stored on disk or the output file from the previous program which is piped to the FIND filter or data that you type in at the keyboard (everything you type after issuing the FIND command until you enter Ctrl-Z).

The string of characters must be surrounded by quotation marks. If the string itself has quotation marks round it, then both sets of quotation marks must be typed.

The string of characters must be entered exactly as it is in the file if the string is to be found. In particular, it must contain the same upper and lower case characters.

The result of the search is displayed on the screen. This can be:

- the lines in which the string of characters was found
- the number of such lines
- the lines in which the string was not found depending on which of the `/V`, `/C` and `/N` options were selected.

If no option is selected, the lines in which the string of characters was found are displayed.

- **To find a given string of characters in one or more disk files**

Form `FIND` *[/V][/C][/N]* `"string"` *[d:][\][path\]filename.filetype*
[d:][\][path\]filename.filetype...]

- **To find a given string of characters in what you type at the keyboard**

Form `FIND` *[/V][/C][/N]* `"string"`

- **To find a given string of characters in the output of another program**

Form *COMMAND* `|` `FIND` *[/V][/C][/N]* `"string"`

Options	/V	Display all lines except those containing the specified string
	/C	Display the number of lines in which the string was found
	/N	Number the lines according to their position in the file

Examples

- *You want to display all the lines in MYFILE.TXT that contain the word computer.*

  If you are interested in including all the variants (eg.microcomputer, computers, etc.) use the command line:

  ```
  FIND "computer" MYFILE.TXT
  ```

  (assuming that the external command FIND is stored either in the default directory or in a directory that MS-DOS automatically searches - (see Section 7.2.4).

  If you want just the word computer, you might use the command line:

  ```
  FIND "computer" MYFILE.TXT
  ```

  Although this would not pick out computer., computer, etc.

- *You want to display and number all the lines in MYFILE.TXT that contain the phrase* `"Best Fit"`*, complete both quote marks. The command line to use is:*

  ```
  FIND/N ""Best Fit"" MYFILE.TXT
  ```

Note: *The use of two sets of quote marks.*

- *You want to display full details of all the files in the current directory, that were last changed on a particular date, for example, 26 September 1987. To do this, use the instruction* `DIR` *to list the directory. Filter this list before it is displayed on the screen by using the FIND command; this will filter out just those lines containing the date 26-09-86. The FIND instruction you want is FIND* `"26-09-87"`*, giving you a complete command line of:*

  ```
  DIR¦FIND "26-09-87"
  ```

FOR

Batch Subcommand

`FOR %%parameter IN (value[value...]) DO command`

FOR is used to execute a number of similar commands.

%%parameter is set to each value listed in the FOR command in turn. This value is then substituted for %%parameter in command and the command that this forms is then executed.

Your PC goes on to execute the next command in the Batch file when the list of values has been exhausted.

Form `FOR %%parameter IN (value[value...]) DO command`
where parameter is a single character.

Notes: Any character may be used as the parameter but digits are not recommended as the parameter in a FOR command because of the risk of confusion between these parameters and the dummy parameters substituted from the command line.

command cannot be another FOR sucommand.

The list of values can include a group of filenames represented by a file name template, created using the wildcards * and ? (see Part 1). In this case MS-DOS takes as a different value each file in the directory with a matching file name.

Paths cannot be given to files mentioned in the FOR command. So all the files included in the list of values must be in the current directory of the drive.

FOR can also be used outside of any Batch file. In this case, replace %%parameter with %parameter.

Examples

- *To make a Batch file, among other things, copy the files OLDDATA.1 and NEWFILE.2 from the default directory to the current directory on Drive C: include the command line:*

```
FOR %%A IN (OLDDATA.1 NEWFILE.2) DO COPY %%A C:
```

This is exactly equivalent to including the lines:

```
COPY OLDDATA.1 C:
COPY NEWFILE.2 C:
```

- *To make the Batch file, first copy to Drive C, all the files with filenames beginning OLD and then all the files with filenames beginning NEW, include the command line:*

```
FOR %%A IN (OLD*.* NEW*.*) DO COPY %%A C:
```

The files will be copied in the order they appear in the default directory.

- *To process three programs FIRST, SECOND AND THIRD one after another, and when these all require the same complex command tail, you might include the following line in your Batch file:*

```
FOR %%A IN (FIRST SECOND THIRD) DO %%A command-tail
```

This is exactly equivalent to the three lines:

```
FIRST     command-tail
SECOND    command-tail
THIRD     command-tail
```

FORMAT

External Command

`FORMAT` *d:[option[option...]]*

Format the disk in the specified drive

The FORMAT command is used to:

- **prepare a new blank disk for storing data and programs**
- **to reclaim an old disk on which the data has been corrupted**

The simplest form of the command formats the disk according to the characteristics of the drive holding the disk. Thus a disk held in a standard floppy disk drive on the PC will be formatted to have 160 tracks (divided between its two sides) and to have 9 sectors per track.

A floppy disk can be formatted:

- to be used as a Startup disk (ie. it can be used immediately after the PC is switched on)
- to have special numbers of tracks and sectors
- only on one side of the disk

depending on which options are selected. As it finishes formatting one disk, FORMAT asks whether you want to format another. Type `Y [ ↵ ]` to format another in the same way: type `N [ ↵ ]` to leave FORMAT.

Exit codes are set at the end of the format to record whether the format operation was successful. These can be used in a IF command (see Section 7.4).

You have to specify which drive will hold the disk to be formatted. This is to safeguard you against accidentally formatting your default drive. Even so, it is a good idea to write protect the disk by opening the write-protect shutter (or covering the write-protect notch) on the disk holding the FORMAT program: even the most careful user can make typing mistakes!

Note that the `FORMAT` command can be used to format Drive C: - indeed this was performed as part of the Hard Disk installation process. Because formatting Drive C: is potentially a very serious business, the format program issues two warnings before carrying out this process. The first warning (only if some data is detected on the Hard Disk) is:

```
Enter current Volume Label for Drive C:
```

You must then type exactly the Volume Label, followed by [↵], otherwise the FORMAT program will abort. The Volume Label is displayed every time a DIR command is performed and can be changed by the LABEL command. It is strongly recommended that you label your Drive C: with, for example, your name in order to identify directory printouts as well as preventing possible accidents with FORMAT. If the Drive has no label, then simply a [↵] is required by FORMAT.

The second warning is:

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

Only enter Y if you are sure that you want to proceed.

In order to create new MS-DOS Startup disks, you should insert Disk 1 (Microsoft MS-DOS) and use the command:

```
FORMAT B: /S
```

Then follow the instructions on the screen for inserting the disk for Drive A and the disk for Drive B.

Form FORMAT *d:[option[option...]]*

Options

/1	●	Format as a single-sided disk
/8	●	Format with 8 usable sectors per track
/B		Reserve space for hidden system files
/N:*n*	●	Format with n sectors per track (n=8,9)
/T:*t*	●	Format with t tracks (t=40,80)
/V		Prompt for a disk label after the format has been completed (the label, which is shown on the screen every time you use the DIR command to display the contents of the disk, can be up to eleven (11) characters long)
/S		Format and prepare for use as a Startup disk by copying the system files from the default drive to the newly formatted disk

● Support of these options varies from PC model to model please consult the relevant section in Part 1 to discover which options are available to your particular PC.

Exit codes

0	Format successful
3	Terminated by user
4	Fatal error
5	Hard (Winchester) disk format abandoned

Notes: If you have the FORMAT external command stored on a floppy disk, write-protect this disk by opening the write-protect shutter (or covering the write-protect notch with the small sticky label) before you type in a FORMAT command.

If you use the **/S** option, this must be the last option specified in your command line. You should also note that the hidden files IO.SYS and MSDOS.SYS, and the COMMAND.COM file are copied to the disk.

If you use the **/S** option with no MS-DOS Startup disk in the default drive, FORMAT will prompt you to insert a 'System' disk before copying the system files to the new disk. Insert a copy of MS-DOS SYSTEM Disk.

You should not use the FORMAT command with any ASSIGNed, JOINed or SUBSTituted drives (see Section 7.1).

If you use the **/B** option, you can put any version of MS-DOS onto the disk. You are not restricted to the version supplied with your PC.

Examples

- *To format a floppy disk to use simply to store programs and data, put the disk holding the FORMAT command in Drive A, make Drive A the default drive (if it is not already) and use the command line:*

```
FORMAT B:
```

If you have a single-drive PC, you will be prompted when you need to insert the disk for Drive B (the floppy disk you want to format). If you have a two-drive system, insert this disk in Drive B (the righthand disk drive).

- *To format a floppy disk for use as a Startup disk, ie. as a disk which loads the MS-DOS operating system immediately after switching on or resetting your PC, put a copy of the PC Disk 1 (MS-DOS Startup and Utilities) in Drive A, make Drive A the default drive (if it is not already) and use the command line:*

```
FORMAT B:/S
```

If you have a single-drive PC, you will be prompted when you need to insert the disk for Drive B (the floppy disk you want to format). If you have a two-drive system, insert this disk in Drive B (the righthand disk drive).

The final stage of the process will copy the IO.SYS. MSDOS.SYS and COMMAND.COM files from the Startup and Utilities disk onto your new disk.

GOTO

`GOTO` *label*

Go to a labelled point in the file

Normally, the commands in a Batch file are obeyed in the order they appear in the Batch file. The GOTO subcommand changes this by directing MS-DOS to go to a labelled point in the Batch file and execute the commands that follow. It can therefore be used:

- to create a loop of commands within the Batch file
- with the IF subcommand, to execute commands only in certain situations

The next command to be executed is the one immediately after the label.

Form `GOTO` *label*

Where *label* is a labelled point in the Batch file.

Notes: The point in the Batch file is marked by a colon (:) followed by *label* on a line on their own. This name may not include tab, space or equals characters.

The label may be up to 40 characters long but the first eight characters must be different from the first eight characters of any other *label* in the file. If the *label* is not found, the rest of the Batch file is abandoned.

You can put as many labels into a Batch file as you like, MS-DOS ignores all lines starting with a colon unless it is looking for a *label*. Lines starting with a colon can therefore be used to space out a Batch file to make it more readable.

Example

Suppose your Batch file contained the label

```
:Stage 2
```

To tell MS-DOS to execute next the command lines that follow this label, insert the following command line at the appropriate place in your Batch file:

```
GOTO Stage 2
```

GRAFTABL

External command

GRAFTABL *[437|850|860|863|865|/status|?]*

Load Graphics Table

Use the GRAFTABL command to display an extended character set when using display adaptors in graphics mode.

A table of data is loaded into memory which defines the additional ASCII characters, 128 to 255.

Form GRAFTABL*[437|850|860|863|865|/STATUS|?]*

Options

437 850 860 863 865	code page numbers:
437	United States
850	Multilingual
860	Portugal
863	Canada
865	Norway and Denmark
/STATUS	displays the number of the selected country code page
?	displays summary of parameters

Example:

To load the table of graphics characters for the Portuguese code page, type:

```
GRAFTABL 860
```

ASCII characters, 128 through to 255, can then be displayed in the graphics modes.

Additional Information

Exit codes The GRAFTABL command returns the following exit codes:

0	Command successful
1	Table already loaded
2	File error occured
3	Incorrect parameter; no action taken.
4	Incorrect version of DOS.

GRAPHICS

External command

`Graphics` *[printer][/b][/p=port][/r]*

Print a graphics display screen on a printer

The GRAPHICS command allows you to print a graphics display screen on a printer, when you are using a colour graphics adaptor.

Form `GRAPHICS` *[printer][/b][/p=port][/r][/lcd]*

where `printer` 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.

Note The printer type default value is **graphics** i.e. **print on an IBM Personal Graphics Printer or IBM Proprinter.**

Options The GRAPHICS command accepts the following switches:

`/b`	Prints the background in colour. This option is valid for COLOR4 and COLOR8 printers.
`/p=port`	Sets the parallel printer port, to which GRAPHICS sends its output, when you press the **[PrtSc]** key. 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 setting is to print black where there is white on the screen; and white where there is black on the screen.

Notes i) To print the screen, press the **[PrtSc]** key.

ii) Only 320 x 200 (4 color) and 640 x 200 (2 color) graphics modes are supported:

- if the computer is in 320 x 200 color graphics mode, and if the printer type is color 1 or graphics, the GRAPHICS command prints the screen contents with up to four shades of grey.
- if the computer is in 640 x 200 color graphics mode the GRAPHICS command prints the screen contents sideways on the paper.

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

Examples

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

```
graphics
```

Then, when the information you want to print is displayed on the screen, press the **[PrtSc]** *key.*

IF *(NOT) condition command*

The IF subcommand is used to make executing a command depend on whether particular statements are true or false.

- **When IF is used, the command is executed if the condition is true.**
- **When IF NOT is used, the command is executed if the condition is false.**

The conditions that can be tested are:

- the value of the exit code set by some programs to record the reason the program finished
- whether a dummy parameter (which may be a parameter representing an evironment string) is set to a particular string of characters
- whether a particular file exists

An IF subcommand with a GOTO subcommand as *command*, can form a conditional loop in the sequence of commands, or conditionally leave out a whole series of commands.

- **Execute command if the condition is true**

Form IF *condition command*

- **Execute command if the condition is false**

Form IF NOT *condition command*

Testable conditions

ERRORLEVEL *n*

Returns TRUE if the previous command set an exit code of *n* or greater: otherwise FALSE.

STRING1==STRING2

Returns TRUE if the two strings of characters, however they are represented in the command line, are identical: otherwise FALSE.

`EXIST` *[d:]filename.filetype*

Returns TRUE if the named file is in the current directory on the named drive: otherwise FALSE.

Notes The total command line may not exceed 127 characters (including spaces).

Of the MS-DOS commands, only BACKUP, FORMAT, GRAFTABL, KEYB, REPLACE, RESTORE and XCOPY return an exit code. However, the utilities described in PART 1 will usually return error codes and your own programs may return exit codes.

The strings that are compared may not include tab, space or equals characters.

The two strings will not be regarded as the same if they have the same characters but do not match where letters are upper and lower case.

Examples

- *To copy the file OLDDATA.1 on the default drive to Drive C at this point in the Batch process only if the previous command had failed (an exit code greater than 0), use the command line:*

```
IF ERRORLEVEL 1 COPY OLDDATA.1 C:
```

- *To copy the file OLDDATA to Drive C only if the previous command has been successful (an exit code of 0), use the command line:*

```
IF NOT ERRORLEVEL 1 COPY OLDDATA.1 C:
```

- *To check that the file NEWFILE.2 exists before copying it to Drive C, use the command line:*

```
IF EXIST NEWFILE.2 COPY NEWFILE.2 C:
```

- *To copy NEWFILE.2 to Drive C and then rename the original file OLDDATA.2 at this point only if NEWFILE.2 exists, you need to divert the processing beyond the command lines to run these programs. You might, therefore, have lines similar to the following in your Batch file:*

```
...
IF NOT EXIST NEWFILE.2 GOTO Diversion
COPY NEWFILE.2 C:
REN NEWFILE.2 OLDDATA.2
:Diversion
...
```

- *To print out the file OLDDATA.1 using a COPY command only if an environment parameter that you have called PRINTER is currently set to the string* `ON`*, your IF command needs to compare the value of the parameter PRINTER with the string* `ON`*. Because PRINTER is an environment parameter, the string it is set to is represented in the IF command by* `"%PRINTER%"`*, making the command line to use:*

```
IF "%PRINTER%"=="ON" COPY OLDDATA.1 PRN
```

JOIN

External command

`JOIN d: (d:)(\)path`

Join a disk drive into another disk's directory structure

Join adds the directory structure on one drive onto the directory structure on a second drive, It does this by telling MS-DOS to interpret any path to a directory on the second drive as an instruction to look on the first drive.

The directory to which the drive is joined, must be empty and it must be in the root directory of the second drive. If the directory you specify in the JOIN statement does not already exist, the JOIN command will create it for you.

If you use command on its own without a command tail, JOIN displays a list of all the joins that are currently in force.

- **To join a drive to a directory**

Form `JOIN` *d:* *(d:)(\)path*

The directory the drive is joined to
The drive joined to the directory

- **To break the join**

Form `JOIN` *d:/D*

The drive that had been joined

- **To display current joins**

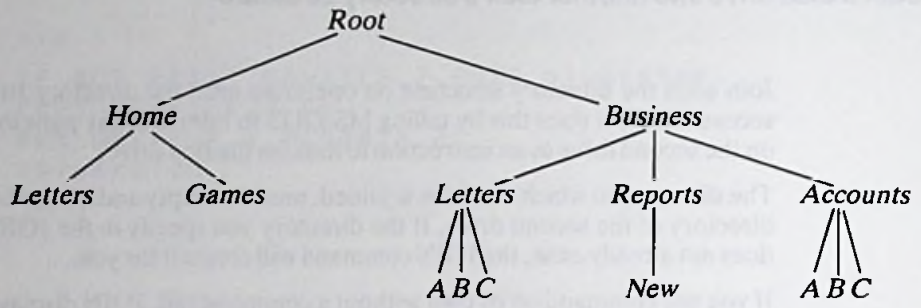
Form `JOIN`

Notes After you have joined a drive to a directory, do not try to use the joined drive in the normal way. MS-DOS now regards this drive as an 'Invalid drive'.

If the directory is not empty when you try to make the join, the message **Directory not empty** will be displayed.

You cannot break the join between a drive and a directory if the file holding the external JOIN is in the drive.

Suppose the directory structure on the disk in Drive A is like this:



- *To put the disk in Drive C, into this directory structure, you would type a command line of the form:*

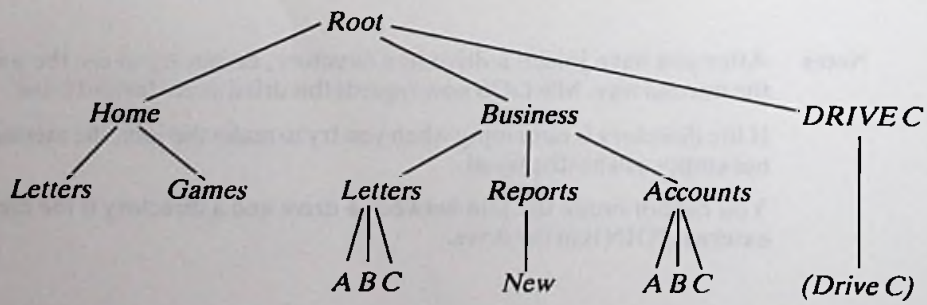
```
JOIN C: A:\path
```

(assuming that the external command JOIN is stored either in the default directory or in a directory that MS-DOS automatically searches - see Section 7.2.4)

- *To make a specially created directory called Drive C represent the drive, use the command line:*

```
JOIN C: A:\DRIVEC
```

This will automatically create the new directory for you and position it in the Root directory, giving you the directory structure:



- *You might want to make Drive C part of the Home directory and so type the command line:*

  ```
  JOIN C: A:\HOME
  ```

  This will be rejected because the directory must be empty. The Home directory has at least two subdirectories in it (ie Letters and Games)

- *Knowing that the directory called New was empty, you might type the command line:*

  ```
  JOIN C: A:\BUSINESS\REPORTS\NEW
  ```

  This would be rejected by MS-DOS. The drive can only be joined to a subdirectory of the Root directory.

- *To break the connection between Drive C and the directory DriveC, use the command line:*

  ```
  JOIN C:/D
  ```

KEYB

External command

KEYB*[xx[,[yyy],[[d:][path]filename[.ext]]]]*

Load a keyboard program

Use KEYB to support all keyboards other than U.S. English keyboards. KEYB loads a keyboard program, replacing the keyboard program resident in ROM BIOS.

For detailed information about code page switching see the command, CHCP, and the section entitled Code Page Switching in Part one.

Form KEYB*[xx[,[yyy],[[drive:][path]filename[.ext]]]]*

where:

xx is a two letter keyboard code

yyy is the code page which defines the character set

[[d:][path]filename[.ext]] specifies the drive, path and filename of the keyboard definition file, KEYBOARD.SYS. If this parameter is omitted, KEYB will look for the file KEYBOARD.SYS in the root directory of the current drive.

Options The above parameter xx is one of the following codes:

Keyboard Type	Code	Command
Belgium	be	keyb be
Denmark	dk	keyb dk
France	fr	keyb fr
Finland	su	keyb su
Germany	gr	keyb gr
Italy	it	keyb it
Latin America	la	keyb la
Netherlands	nl	keyb nl
Norway	no	keyb no

Portugal	po	keyb po
Spain	sp	keyb sp
Swiss-French	sf	keyb sf
Swiss-German	sg	keyb sg
Sweden	sv	keyb sv
United Kingdom	uk	keyb uk
United States	us	keyb us

If you type KEYB, without options, MS-DOS will display a message like this:

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

which shows the current keyboard code, its related code page and the current code page used by your console screen device (CON).

The available Code Pages are as follows:

VALUE	CODE PAGE
437	United States
850	Multilingual
860	Portuguese
863	French-Canadian
865	Nordic

Notes From the KEYB program, you can switch from the current keyboard to the default keyboard, by holding down the **[Ctrl]** and **[Alt]** keys and pressing the **[F1]** key.

To switch back again, hold down the **[Ctrl]** and **[Alt]** keys and press the **[F2]** key.

The KEYB program returns the following exit codes:

EXIT	CODES
0	Successful execution and termination
1	Invalid language, code page, or syntax
3	KEYB could not create a keyboard table in resident memory
4	An error condition occured when communicating with the CON device
5	Code page request has not been prepared
6	The translation table for the selected code page cannot be found in the resident keyboard table

KEYBOARD.SYS

The KEYBOARD.SYS file contains tables which direct the KEYB.COM command to convert scan codes to ASCII characters. To change a keyboard, the new keyboard must support at least one of the currently prepared code pages for the CON device.

The MODE command is used to prepare the new code pages required by the new keyboard layout; only certain keyboards or code page combinations are allowed. If you do not match the correct keyboard and display, character keys may not be correctly translated in the respective code page. Incorrect characters may be displayed as a result.

LABEL

External command

LABEL *[d:]label*

Create or change a disk label

The LABEL command is used to give your disks unique labels which you can thereafter quickly check when you want to find out which disk you have in a drive. The label is always displayed when you use a DIR command, even when you are looking at a directory which is low down in the directory tree. Disk labels can also help in cataloguing your disks.

The label can be up to 11 characters long, including spaces.

If you do not type in a new label, MS-DOS displays the existing label (if any) and asks you what new label you want to give the disk. Just press the [←] key either to keep the existing label or to delete the label. (You will be asked which of these options you require).

Form LABEL *[d:][label]*

Notes You can use any characters in the disk label except: * ? / \ | . , ; : + = < > [ ]

Spaces are allowed. If you try to use any of these characters, your new label will not be accepted and you will see an error message.

You can't label a disk if the drive it is in has been ASSIGNed to another drive or JOINed to a directory.

If you specify more than 11 characters, the first 11 are used as the label and all subsequent ones are ignored.

Example

To give the disk in Drive B the label RECORDS 86-7, either use the command line:

```
LABEL B:RECORDS86-7
```

which does the whole job, or just:

```
LABEL B:
```

Then you will see something like this on the screen:

```
A>label b:
Volume in drive B label RECORDS
Volume label (11 characters, ENTER for none)?
```

To set the new label type:

```
RECORDS86-7
```

Both these commands assume that the external command, LABEL, is stored either in the default directory or in a directory on the search path - see section 7.2.4.

MKDIR

Internal command

`MKDIR` *[d:][\]path*

Make a new directory

MKDIR adds a new directory to the directory structure on a particular DOS disk. The path to the new directory is defined through existing directories either from the disk's root directory or from the drive's current directory. The new directory is attached to the existing directory specified by all but the last section of the path.

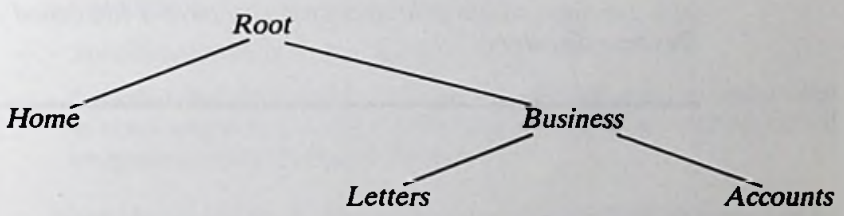
MKDIR can be abbreviated to MD

Form `MKDIR` *[d:][\]path*
or `MD` *[d:][\]path*

Notes Include the optional backslash if the path starts at the root directory. If the path starts from the current directory, leave this backslash out.

Examples

Suppose you wanted to create the following pattern of directories on a new disk, which you have placed in the default drive:



- *To create the Home and Business directories (both subdirectories of the Root directory) you would use the command lines:*

```
MD \HOME
MD \BUSINESS
```

- *To create the Letters and Accounts directories - subdirectories of the Business directory, you could either use the command lines:*

```
MD \BUSINESS\LETTERS
MD \BUSINESS\ACCOUNTS
```

or you could change the current directory to the BUSINESS directory and then create its subdirectories, with the command lines:

```
CD \BUSINESS
MD LETTERS
MD ACCOUNTS
```

Notice that you do not put a backslash in front of LETTERS or ACCOUNTS, to show that these are to be subdirectories of the current directory. If you had put a backslash, the new directories would have been formed in the Root directory.

- *If, at a later date, you forget that you have created the Accounts directory, you might type the command:*

```
MD \BUSINESS\ACCOUNTS
```

Because this directory already exists, you will see the message:

```
Unable to create directory
```

This message would also appear if you have a file called ACCOUNTS in the Business directory.

MODE

`MODE` *device*`:`*setting*`[,`*setting...*`]`

Set Mode of Operation for Input and Output devices

The MODE command is used to:

- **Prepare MS-DOS for communication with parallel printers**
- **Prepare MS-DOS for communication with asynchronous devices**
- **Prepare parallel printers and console screen devices for code page switching (See the section, Code Page Switching, in Part One, for further information)**
- **Redirect output**

- **Parallel printer modes**

Form `MODE LPT`*n*`[:][`*chars*`][,[`*lines*`][,P]]`

where:

n - Specifies the printer number: 1,2 or 3

chars - Specifies characters per line: 80 or 132

lines - Specifies vertical spacing, lines per inch: 6 or 8

P - Specifies that if a timeout error occurs, MODE will try continuously to send output to the printer. This option causes part of the MODE program to remain resident in memory

Note: do not use the P switch when using the MODE command over a network.

The default settings are:

- LPT1
- 80 characters per line
- 6 lines per inch

Notes i) To break out of a time-out loop, press the **[Ctrl]** and **[Break]** keys.

ii) For parallel printer modes, you can use PRN and LPT1 interchangeably.

Examples

If you want to print on a parallel printer that is connected to your computer's first parallel printer port (LPT1), with 80 characters per line and 8 characters per inch, type:

```
MODE LPT1:80,8
```

or

```
MODE LPT1:,8
```

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

```
MODE LPT1:80,8,p
```

If you want the computer to stop continuously retrying to print, press the **[Ctrl]** *and* **[Break]** *keys, or type the MODE command without the P option.*

● **Asynchronous communications modes**

Use the following form of the MODE command to prepare MS-DOS for communication with serial printers, modems and console screens.

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

where:

m	- Specifies the asynchronous communications (COM) port number: 1, 2, 3 or 4
baud	- Specifies the first two digits of the transmission rate: 110, 150, 300, 600, 1200, 2400, 4800, 9600 or 19200 i.e. 11, 15, 30 are specified
parity	- Specifies the parity: N (none), O (odd), or E (even) The default value is E

databits - Specifies the number of data bits: 7 or 8 The default value is 7

stopbits - Specifies the number of stop bits: 1 or 2. If baud is 110, the default value is 2. Otherwise the default value is 1

P - Specifies that MODE is using the COM port for a serial printer and that it is will continuously retry to send output to the printer if time out errors occur

Note: do not use the P switch if you are using the MODE command over a network

The default settings are:

- COM1
- even parity
- 7 databits

Example

To set the mode of operation for port 1 to 1200 baud rate, no parity, 8 databits and 1 stopbit, type:

```
MODE COM1:12,n,8,1
```

● **Display modes**

Use the following form to set the MODE command for use with a display:

Form `MODE` *display*

or

`MODE` *[display],shift[t]*

where:

display Specifies one of the following values: 40, 80, BW40, BW80, CO40, CO80 or MONO

- 40 and 80 indicate the number of characters per line

- BW and CO refer to a color graphics monitor adaptor with color disabled (BW) or enabled (CO)

- MONO specifies a monochrome display adaptor with a constant display width of 80 characters per line

shift — Specifies whether to shift the display to the left or to the right

Valid values are:

L - for left

R - for right

t — Tells MS-DOS to display a test pattern in order to align the display on the screen

To improve readability, you can shift a display to a color/graphics monitor adaptor: 1 character (for 40 columns) or 2 characters (for 80 columns in either direction. If you specify t in the MODE command, and a prompt will appear you will be asked whether the screen is aligned properly. If you type Y, the command will end. If you type N, the shift is repeated, followed by the same prompt.

Example

If you type the following command line:

```
MODE 80,r,t
```

the mode of operation will be set to 80 characters per line and shifts the display 2 characters to the right. The test pattern will be displayed, allowing the opportunity to further shift the display without having to enter the command again.

Note *Shifting the display causes all MODE resident code to be loaded.*

- **Device code page modes**

  You can use the MODE command to set or display code pages for parallel printers or your console screen device.

Forms `MODE` *device* `CODEPAGE PREPARE=`*((cplist)[d:][path]filename[.ext])*

and

`MODE` *device* `CODEPAGE SELECT=`*cp*

`MODE` *device* `CODEPAGE REFRESH`

`MODE` *device* `CODEPAGE`*[/status]*

where:

device — Specifies the device to support code page switching

Valid device names are:

- CON
- PRN
- LPT1
- LPT2
- LPT3

cp/cplist — Specifies a code page or lists more than one code page, separated by commas or spaces.

Valid code pages are:

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

filename — Identifies the name of the Code Page Information (.cpi) file that MS-DOS should use to prepare a code page for the specified device

Keywords

There are four keywords that you can use with the MODE device command.

Each keyword makes the MODE command perform a different function:

- PREPARE — *Tells MS-DOS to prepare code pages for a given device*

  You must prepare a code page for use with a device
  PREPARE can be abbreviated to PREP

Example

To prepare your display to use code page 437, type:

```
MODE CON CP PREPARE=((437) C:\DOS\EGA.CPI)
```

To prepare your printer to use code pages 850 and 865, type:

```
MODE LPT1 CP PREPARE=((850,865) C:\DOS\4201.CPI)
```

- SELECT — *Specifies which code page you want to use with a device*

  You must PREPARE a code page before you SELECT it
  Select can be abbreviated to SEL

Example

Having prepared code page 850, you can then use the MODE SELECT command to identify the code page to be activated. Type:

```
MODE LPT1: CP SELECT=850
```

- REFRESH — *This keyword will reinstate prepared code pages that have been lost due to hardware, or other errors Refresh can be abbreviated to REF*

Example

If you are using code page 860, and the printer is switched off accidently, it will return to its default code page. To make the printer regain the lost code page, type:

```
MODE LPT1 CP PREP=((860) C:\DOS\4201.CPI)
```

or

```
MODE LPT1 CP REFRESH
```

- /STATUS

  Display the current code pages prepared and/or selected for a device

  Typing **/STATUS** *is optional*

  The following commands both produce the same results:

  ```
  MODE CON CODEPAGE
  MODE CON CODEPAGE/STATUS
  ```

  /STATUS can be abbreviated to /STA CODEPAGE can be abbreviated to CP

- **Redirecting output**

  If you want your computer to send its printer output to a serial printer, you will have to use the MODE command twice.

  The first MODE command specifies the asynchronous communications modes, and the second MODE command redirects the computer's parallel printer output to the asynchronous communication port, specified in the first MODE command.

Example

If your serial printer operates at 4800 baud with even parity, and it is connected to the COM1 port, type:

```
MODE COM1:48,E,,,P
MODE LPT1:=COM1:
```

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

```
MODE LPT1
```

This command will disable any redirection of LPT1.

MORE

MORE

Display output one screenful at a time

The MORE filter is used to control the display of the program output on the screen so that this is only displayed one screenful at a time. The next screenful will only be displayed after you have pressed the key, and so on until all the output has been displayed.

Form *command* ¦ MORE

Example

Your program ANALYSIS.EXE displays its results on the screen. To see the results of processing the data stored in MYDATA.RAW a page at a time, use the command line:

```
ANALYSIS MYDATA.RAW¦MORE
```

(assuming that the external filter MORE is stored either in the default directory or in a directory that MS-DOS automatically searches - see Section 7.2.4)

Press [↵] *to see the next page of output.*

NLSFUNC

External command

NLSFUNC*[[d:][path]filename]*

Load country-specific information

NLSFUNC supports the use of extended country-specific information and code page switching. See the CHCP command and the Code Page Switching section, in Part one for more information.

Form NLSFUNC*[[d:][path]filename]*

where: filename specifies the file containing country-specific information.

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

Example:

Loading country-specific information:

If you want to use country-specific information from a file called, for example, GERCDPG.SYS, *rather than from the COUNTRY.SYS file, type:*

```
NLSFUNC GERCDPG.SYS
```

The default country-specific information is to be found in the COUNTRY.SYS file. To use this type:

```
NLSFUNC
```

PATH

Internal command

`PATH` *[d:]\path[;[d:]\path...]*

Define a search path

Normally, if you do not say which directory a program file is in, MS-DOS tries to find it in the default directory. PATH sets up a sequence of directories which MS-DOS will search, for program files, alongside the default directory. In effect, what happens is that each path given in the PATH command is applied to the program filename in turn until the file is found. Each path specifies a separate directory. The default directory is always searched first.

This gives you a neat way of, for example, ensuring external commands are accessible without clogging up your current directory.

- **To set up a new path**

Form `PATH` *[d:]\path[;[d:]\path...]*

- **To clear the existing path**

Form `PATH;`

Notes Each path must start from a root directory.

The directories are searched one by one in the order given in the PATH command until a file matching the file specification is found or the sequence of directories is exhausted.

You must keep the search path up to date. If MS-DOS tries to search a directory that no longer exists, your command will fail.

Examples

- *You have stored all the MS-DOS external commands in a directory called* \COMMANDS on Drive C. To get MS-DOS to search for the external commands as if they were always in the default directory, use the command line:

```
PATH C:\COMMANDS
```

- *The current search path is:*

  ```
  A:\MSDOS;A:\WP
  ```

  and you want to extend this list to include C:\COMMANDS. The command line to use is:

  ```
  PATH A:\MSDOS;A:\WP;C:\COMMANDS
  ```

- *You delete the \COMMANDS directory which you know is on the current search path. To bring the path up to date, first discover the current search path by typing the command line:*

  ```
  PATH
  ```

  If the response is:

  ```
  A:\MSDOS;A:\WP;C:\COMMANDS
  ```

  the command line you need to type is:

  ```
  PATH A:\MSDOS;A:\WP
  ```

- *To clear the path, thus making the default directory the only directory to be searched, use the command line:*

  ```
  PATH;
  ```

PAUSE

Batch subcommand

PAUSE *[remark]*

Pause while processing

PAUSE is used to halt the execution of a Batch file, so that you can change disks for example. The message `Press any key when ready...` is automatically displayed.

The remark will not be displayed if the ECHO subcommand has been used to turn off the display of command lines on the screen. However, the standard message continues to be displayed. Use ECHO to display the message in this case.

Form PAUSE *[remark]*

Notes To continue execution of the Batch file, press any character key - for example, the Space bar. To abandon it , press either Ctrl- Break or Ctrl-C.

The total length of the command line can be up to 127 characters long. There is no restriction on the characters that can be used in the remark.

The remark will not be displayed if the ECHO subcommand has been used to turn off the display of command lines on the screen. However, the standard message continues to be displayed,. Use ECHO to display the message in this case.

Example

At the point in a Batch file at which you need to exchange the disk currently in disk drive for the disk you know as the Records Disk, you might include the command line:

```
PAUSE Insert Records Disk now
```

When the Batch file is processed, what you see on the screen will be:

```
PAUSE Insert Records Disk now
Press any key when ready...
```

The second part of this message is always displayed - whatever message you set up. Press a character key, such as the Space bar, when you are ready for the Batch process to continue.

PRINT

External command

PRINT *[d:][\][path\]filename.filetype[[d:][\][path\]filename.filetype...][/P][/C][/T]*

Print text files in the background

The PRINT command is used to print one or more text files on a line printer attached to your PC at the same time as you do other work. With the /P,/C, and /T options, the command can also be used to:

- add files to the list of those currently waiting to be printed
- remove files from this list
- abandon printing the remaining files in the list

You can use a wildcard file name template to insert or remove a number of files from the print queue. Files inserted in the list in this way will be printed in the order they appear in the directory.

If the PRINT command is used on its own, without a command tail, a list of all the files waiting to be printed is displayed.

The first time you use the PRINT command after you switch on or reset your PC, there are a number of other parameters which you can set including:

- the device name your printer has (see Section 4.1)
- the size of the print buffer
- how many files can be waiting to be printed
- how much of your PC's processing time is given over to printing the files

NOTES The disk containing the files to be printed must remain in the specified drive until printing is completed. Files waiting to be printed may not be modified until after they have been printed.

PRINT takes over the printer entirely. Any attempt, for example, to echo output sent to the screen on the printer or to produce screen images on the printer will just produce an `Out of paper` message until all the files in the print queue have been printed.

If PRINT encounters a disk error in reading a file, the file will be abandoned. PRINT then goes on to printing the next file in the queue.

Tab characters in the files are replaced by spaces up to the start of the next 8-character column.

The first time you use the PRINT command

Form `PRINT` *[d:][\][path\]filename.filetype[[d:][\][path]filename.filetype...][option[option...]]*

File or files to be printed *Set-up options*

Set-up options

`/D:`*device* Specifies the device name used for your printer. (The default is PRN) **Note:** this option must be listed first.

If you do not include this option, MS-DOS displays the message `Name of list device [PRN]:` and gives you the opportunity to specify a different output device, If you have a parallel printer, just press the key. If you have a serial printer, give the device name `AUX`.

`/B:`*buffer-size* Sets the size of the print buffer. The default is 512 bytes. As a general rule, a bigger buffer improves the performance of the PRINT command.

`/Q:`*queue-size* Sets the number of files you can have waiting to be printed (between 1 and 32). The default is 10 files.

`/S:`*time-share* Sets how much computer time is devoted to your foreground task compared with that used on printing in the background. The value you set must be between 1 and 255.

The default is 8, ie your PC devotes eight times as much effort to programs you use while the files are being printed, as it does to printing the files. As a result, your files will take roughly nine times longer to print than they would have done if printing was the only job your PC was doing.

`/U:`*busyticks* Sets how long your PC will wait for the printer to become ready to receive more data to print before going back to working on your foreground task. The default is 1 processor clock-cycle.

`/M:`*maxticks* Sets the maximum length number of processor clock-cycles your PC will spend sending data to the printer before going back to working on your foreground task (between 1 and 255). The default is 2 clock-cycles.

Notes Be careful only to name an output device attached to your PC as the output device for the PRINT command.

Once the PRINT command has been used, the amount of memory available for running your programs is reduced until the next time you reset your PC.

If you try to use any of the set-up PRINT options in a subsequent PRINT command, the command will fail and MS-DOS will display the message Invalid parameters.

Subsequent occasions

- **To change the list of files to be printed:**

Form `PRINT` *[d:][\][path\]filename.filetype* `/P | /C[[d:][\][path\]` *filename.filetype*`[/P | /C]...]`

Options

`/C` Remove this and following files (up to the next file listed with the `/P` option) from the list of files waiting to be printed.

`/P` Add this and following files (up to the next file listed with the `/C` option) to the list of files waiting to be printed.

- **To abandon printing the current list**

Form `PRINT /T`

Note If a file is currently being printed when this command is being entered, a cancellation message will be printed. The paper is then advanced to the next page.

Examples

- *To print the file MYFILE.TXT on the default drive, accepting all the default settings, use the command line:*

```
PRINT MYFILE.TXT
```

(assuming that the external command PRINT is stored either in the default directory or in a directory that MS-DOS automatically searches - see Section 7.2.4)

If this is the first time you have used the PRINT command since seitching on or resetting your PC, the following message will appear on your screen:

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

This message will not appear if you have already used a PRINT command.

If you have a parallel printer, just press . If you have a serial printer attached to the Serial Interface, type AUX or COM1 before pressing .

- *To set up a print queue of all your .TXT files (12 files in all) as your first PRINT job since switching on, use the command line:*

```
PRINT *.TXT/Q:12
```

You need the /Q option because otherwise PRINT will cut short your print queue at 10 files.

*If you have already used a PRINT command but with the standard print queue, you will need to reset your PC (**[Ctrl] [Alt] [Del]**) before using this command line.*

You must not change the disk in the default drive while these files are being printed, nor can you modify any file before it has been printed, but you can do other work while the files are being printed.

- *To abandon all the print queue after printing has started use the command line:*

```
PRINT /T
```

- *If, after you have entered the command to print all your .TXT files, you decide that you did not want to print the file MYFILE.TXT, you can remove this file from the list with the command line:*

```
PRINT MYFILE.TXT/C
```

If you did not want to print any file that matched the file name template MY.TXT, you remove all these files with the command line:*

```
PRINT MY*.TXT/C
```

If one of these files was being printed when you typed this command line, a cancellation message will be printed before PRINT goes on to print the next file in the list.

- *If, after starting printing, you decide to add the file MYFILE.BAS to the list of files to be printed, use the command line:*

```
PRINT MYFILE.BAS/P
```

This file will be added to the list, provided there is room in the print queue for this additional file.

*If you decide both to remove all the MY*TXT files from the print queue and add MYFILE.BAS, you can do this with the command line:*

```
PRINT MY*.TXT/C MYFILE.BAS/P
```

Be careful to include the `/P` *at the end of the command line or MS-DOS will try to remove MYFILE.BAS from the current print queue.*

PRINTER.SYS

Device driver

DEVICE=*[d:][path]printer.sys lptx=[type[,hwcp[,...]][,n]]*

PRINTER.SYS is a printer driver which provides code page switching support for parallel ports LPT1, LPT2, and LPT3.

For information about code page switching see Part One, 'Code Page Switching'.

Installation

To install PRINTER.SYS, insert a command line of the following form in your CONFIG.SYS file:

Form DEVICE=*[d:][path]printer.sys lptx=[type[,hwcp[,...]][,n]]*

Options The PRINTER.SYS device driver accepts the following options:

OPTION	FUNCTION
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.

PROMPT

Internal command

PROMPT *prompt-text*

Set up a new system prompt

PROMPT is used to set up a new system prompt message. This can contain simple text and/or information such as the date, the time and the MS-DOS version number as well as, or instead of, the standard details (the default drive). The new prompt is then used until the computer is reset.

The position of any special information you want to include in the prompt text (eg. the date) is marked by a $ followed by a particular character. This representation is also used for some special characters. including $ itself.

If no prompt text is given, by the system prompt goes back to the standard prompt - the default drive followed by a >

Form PROMPT *[prompt-text]*

where *prompt-text* is: *letter|*$*character[letter|*$*character...]*

$*character* is used to represent an information string or a special character as in the table below.

$t	Time
$d	Date
$n	The drive letter of the default drive
$p	The current directory on the default drive
$v	The MS-DOS version number
$$	A $ character
$g	A > character
$l	A < character
$b	A ¦ character
$q	A = character
$h	A Backspace character
$e	An ESCape character
$_	A new line

Any other characters after a $ are ignored.

Example

If you would like the system prompt to have the following form:

Date Time (hours and minutes)
Default-drive Current-directory>

You will need:

`$d`	*To give you the date*
A couple of spaces	*To separate the date from the time*
`$t`	*To give you the time*
`$h$h$h$h$h$h`	*To backspace over the seconds and hundredths of a second in the time*
`$_`	*To give you a new line*
`$n`	*To give the default drive*
`:`	*To give a colon character*
`$p`	*To give the current directory*
`$g`	*To give a > character.*

The command line to use is therefore:

```
PROMPT $d    $t$h$h$h$h$h$h$_$n:$p$g
```

The system prompt will then look something like this:

```
Wed 2-01-1980          22:55
C:C:\>
```

RECOVER

External command

`RECOVER [d:][path]filename / recover[d:]`

Recover a file or a disk containing bad sectors

You can use the CHKDSK command to find out if a sector on your disk is bad, and then the RECOVER command to recover the entire disk or just the file containing the bad sector. MS-DOS will read the file, sector by sector, skipping the bad sectors. When MS-DOS finds a bad sector, it will no longer allocate your data to that sector.

Form `RECOVER [d:][path\]filename`

or `RECOVER [d:]`

WARNING

RECOVER will rename all files it encounters, changing both filename and filetype. A disk in such condition is almost impossible to work with. It is therefore recommended that RECOVER is only used by experts in the most extreme situations.

Example:

To recover a disk in drive A, type:

```
recover a:
```

To recover a file named purchase.1, that has a few bad sectors, type:

```
recover purchase.1
```

Notes: *i) RECOVER does not work on a network from a remote work station.*

ii) RECOVER does not work on drives used in the SUBST or JOIN commands.

REM

REM *[remark]*

Display remark

REM marks the point in the Batch file a which you want a message to be displayed on the screen and lets you set the text of that message. This text is displayed when the Batch process reaches the REM command.

Form **REM** *[remark]*

Notes The total length of the command line can be up to 127 characters long. There is no restriction on the characters that can be used in the remark.

The remark is not displayed if the ECHO subcommand has been used to turn off the display of command lines on the screen. Use ECHO to display the remark in this case.

The special action of characters <> and % is still performed, even when included in a remark.

Examples

- *To arrange for the message Stage 1 finished; Stage 2 started to appear on the screen at a particular point in the Batch process, you could insert at this point the command line:*

```
REM Stage 1 finished; Stage 2 started
```

However, this message will not appear on the screen if you have previously used an ECHO OFF command. If you still want to see the message, you should use an ECHO command rather than a REM command.

- *To make the Batch file more readable when you display it, you might make this part of the file:*

```
...
REM
REM Stage 1 finished; Stage 2 started
REM
...
```

Note how REM commands have been used to provide space around the 'real' REM command.

RENAME

Internal command

`RENAME` *[d:][\][path\]old-name new name*

Rename a file

RENAME is used to change the name of a file or a group of files. Renaming a file does not change its location on disk.

RENAME can be shortened to REN.

Note: You cannot rename a directory. The only alternative is to create a directory with the new name and copy all the files in the current directory to the new directory.

Form
or

`RENAME` *[d:][\][path\]old-name new-name*
`REN` *[d:][\][path\]old-name new-name*

where *old-name* is: *filename*.*filetype*
new-name is: *filename*.*filetype*

Notes The wildcard characters * and ? may be used to specify a group of files in the same directory. However, their use in giving the new file template must be identical to their use in giving the old file template.

If a file already exists with the same file name as the new file name, then renaming will be abandoned. The message `File not found` will be displayed on the same screen.

Examples

- *To rename the file NEWDATA.1 in the default directory OLDDATA.1, use the command line:*

```
REN NEWDATA.1 OLDDATA.1
```

If the file was in the DIR directory on Drive C (a subdirectory of the Root directory), the command line would be:

```
REN C:\DIR1\NEWDATA.1 OLDDATA.1
```

Notice that you do not have to repeat the drive and directory information.

- *To rename all the MYFILE* files in the default directory YOURFILE*, use the command line:*

```
REN MYFILE.* YOURFILE.*
```

*The use of the wildcards * and ? in the new names must be identical to its use in the old names.*

REPLACE

External command

REPLACE *source-drive:[\][path\]filename.filetype target-drive:*
[\]path][option[option...]]

Replace old version and insert new files

The REPLACE command is used to keep back-up disks up to date by:

- **replacing old files on your back-up disk with the newest versions of the files**
- **adding to your back-up disk files that did not exist the last time you brought your back-up disk up to date**

The standard REPLACE command looks for Read-Write files in the target directory which match the file name template given in the command line. When it finds one which also has the same name as a file in the source directory. REPLACE replaces this file with the version from the source disk. The options allow you to:

- add any new files to the back-up disk
- replace Read-Only files as well as Read-Write files
- only replace a file if the source file is newer than the version on the target disk
- search subdirectories on the target disk for matching files
- confirm file-by-file that the old version is to be replaced

If an error occurs during the insert, REPLACE sets an exit code which can be used in an IF command (see Section 7.4).

Form REPLACE *[source-drive:][\][path\]filename.filetype target-drive:[\][path][*
option[option...]]

where filename.filetype is a template representing a number of files

Options

/A	Add new files to the target directory (Cannot be used with the /S option)
/P	Prompt for confirmation before replacing a target file or adding a new source file
/R	Replace both Read-Write and Read-Only files
/S	Search both the given target directory and its subdirectories for old versions of the files (Cannot be used with the /A option)
/W	Wait until a key is pressed before starting to replace and add files (ie. allows you to swap the disks in your floppy disk drives after issuing the instruction)

Notes If you do not specify the `/R` option, any attempt to replace Read-Only files will cause the insert process to stop.

If you do not specify the `/W` option, the process of replacing and adding files, will start working immediately, with the disks currently in the drive(s).

You cannot use REPLACE to update the hidden files. Use the SYS command to do this (see Chapter 5).

REPLACE can return standard MS-DOS error codes as well as specific error codes listed below.

Additional Information

Exit codes

1	Command line error, eg. incompatible options
2	No matching source or target files were found
3	Either the source or the target path was not found
5	Access denied ie. you tried to update a write-protected file
8	Insufficient memory for REPLACE to operate
15	You specified a drive that does not exist

Examples

- *To replace the existing back-up copies on Drive B of .TXT files on the default drive, use the command line:*

```
REPLACE *.TXT B:
```

(assuming the external command REPLACE is stored either in the default directory or in a directory that is automatically searched by MS-DOS - see Section 7.2.4)

This will replace each old version with the new version provided the old version allows Read-Write access. If you want to replace Read-Only files as well, you should use the command line:

```
REPLACE *.TXT B:/R
```

- *If you want to have new versions of all the .TXT files on the default drive on the disk in Drive B - ie. including copies of any new .TXT files that you have created, you should use the command line:*

```
REPLACE *.TXT B:/A[/R]
```

(Use the /R option if you want to replace any Read-Only files on the Drive B disk.)

If you want REPLACE to ask about each file before it copies the new version to your back-up disk, you need to use the /P option - ie. a command line like:

```
REPLACE *.TXT B:/A/P
```

RESTORE

External command

RESTORE *drive1*:*[drive2*:*]filename[*/s*][*/p*][*/b*:*date*][*/a*:*date*]
[/e*:*time*][*/L*:*time*][*/m*][*/n*]

Restore backed up files

Use the RESTORE command to restore files that were backed up using the BACKUP program. RESTORE can restore files from similar or dissimilar disk types, for example, from floppy disk to hard disk.

Form restore *drive1*:*[drive2*:*]filename[*/s*][*/p*] *[*/b*:*date*][*/a*:*date*]*
[/e*:*time*][*/L*:*time*][*/m*] *[*/n*]*

where:

drive1	contains the backed up files
drive2	is the target drive
pathname	identifies the file(s) you want to restore

Options The RESTORE command accepts the following switches:

/s	Restores subdirectories
/p	Prompts for permission to restore any files matching the file specification that are read-only or that have changed since the last backup.
/b:date	Restores only those files last modified on or before date.
/a:date	Restores only those files last modified on or after date.
/e:time	Restores only those 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.

Notes i) RESTORE cannot restore the system files. Use the SYS command to restore these files.

ii) The MS-DOS 3.3 RESTORE command will restore files backed up with either the MS-DOS 3.3 BACKUP commmand, or an earlier version of backup.

Example

To restore the file MANUAL.APP from the backup disk in drive A, to the subdirectory WP on drive C, type:

```
restore a: c:\wp\manual.app
```

Additional information

EXIT CODES: Once the command has been completed, RESTORE returns one of the following exit codes:

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

RMDIR

Internal command

`RMDIR` *[d:][\]path*

Remove a directory

RMDIR is used to delete a directory from the directory structure of a disk.

RMDIR can be abbreviated to RD.

Form `RMDIR` *[d:][\]path*

or `RD` *[d:][\]path*

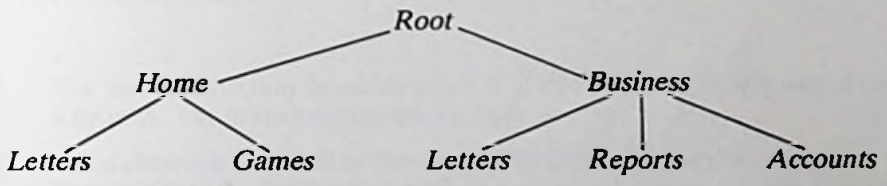
Notes The subdirectory will not be deleted if it contains any files or if it has any references to directories other than the `\.` (current) and `\..` (parent) directories.

The subdirectory will not be deleted if it is the current directory on any drive or if it is assigned to an imaginary drive (see Section 7.1).

Include the optional backslash if the path starts at a root directory; do not include it if the path starts at a current directory.

Examples

Suppose the pattern of directories on the disk in the default was like this:



- *To remove the Reports directory from this directory structure, use the command line:*

```
RD \BUSINESS\REPORTS
```

or, if BUSINESS is the current directory, use the command line:

```
RD REPORTS
```

Remember to clear the Reports directory of all files before using this command; otherwise you will see the message:

```
Invalid directory
```

- *To remove the "Home" part of this directory structure, you need to clear and remove both the \HOME\LETTERS and the \HOME\GAMES directories and only then clear and remove the \HOME directory itself.*

  The first thing to do is to copy every file from all three directories you want to keep to another disk and or another directory. When that has been completed, the command lines to use are:

```
CD \HOME
DEL GAMES\
DEL LETTERS\          - Clear and remove the Games and Letters
RD GAMES                directories
RD LETTERS
CD \                  - Change to a different current directory (you
                        cannot delete the current directory)

DEL HOME\
RD HOME               - Clear and remove the Home directory
```

SELECT

External command

SELECT*[[drive1:] [drive2:][path]][xxx][yy]*

Install MS-DOS on a new floppy disk, or hard disk, with desired country-specific information and keyboard layout

The SELECT command allows you to install MS-DOS on a new disk, with the keyboard layout, and date and time format of the country of your choice.

CAUTION

SELECT should only be used on new disks: the FORMAT command is used in SELECT, therefore, all data on the target disk is destroyed.

The SELECT command does the following:

- Formats the target disk
- Creates both the CONFIG.SYS and AUTOEXEC.BAT files on the new disk
- Copies the contents of the source disk track by track, to the target disk

Form SELECT*[[drive1:] [drive2:][path]][yyy][xx]*

where:
drive 1 is the source drive
drive 2 is the target drive

Notes

i) The source drive may be either drive A or drive B. The default source drive is drive A. The default target drive is drive B.

ii) If you choose a hard disk as the target, MS-DOS will prompt you to type the correct internal label for that disk. If you type the correct label, SELECT displays a warning like this:

```
WARNING, ALL DATA ON NON-REMOVABLE DISK
DRIVE C: WILL BE LOST!
Proceed with format (Y/N)?
```

Options

xxx specifies the country code.

MS-DOS gathers country-specific information, such as time and date formats, from the country.sys file for the specified country code.

yy specifies the keyboard code for the keyboard layout used.

Country and Keyboard Codes

Country	Country code	Keyboard Code
Arabic	785	
Australia	061	US
Belgium	032	BE
Canada (English)	001	US
Canada (French)	002	CF
Denmark	045	DK
Finland	358	SU
France	033	FR
Germany	049	GR
Hebrew	972	
Italy	039	IT
Latin America	003	LA
Netherlands	031	NL
Norway	047	NO
Portugal	351	PO
Spain	034	SP
Sweden	046	SV
Switzerland (French)	041	SF
Switzerland (German)	041	SG
United Kingdom	044	UK
United States	001	US

Example

To create a new MS-DOS floppy disk which includes the country specific information and keyboard layout for Germany:

put the source disk in drive B:

put the target disk in drive A: and type:

```
select b: a: Ø49 gr
```

This message will be displayed:

```
SELECT is used to install DOS the first time.
SELECT  erases everything on the specified
target and then installs DOS.

Do you want to continue (Y/N)? Y
```

If there is data on the disk in drive A and you type Y *,it will be erased.*

Example

To install (or re-install) on a hard disk (drive C):

Put the source disk (MS-DOS System disk) in drive A: and type:

```
select a: c: Ø49 gr
```

This message will be displayed:

```
SELECT is used to install DOS the first time.
SELECT erases everything on the specified
target and then installs DOS.

Do you want to continue?  (Y/N)?  Y
```

If there is data on the disk in drive C and you type Y *it will be erased.*

SET

Internal command

SET *parameter=string*

Set an environment string

The SET command is used to set an environment string, that is one of the special strings of characters that are available to each program you run. The program will respond to this string if it is designed to use the parameter to which it has been set. Environment strings can also be used by Batch processes (see Section 7.4).

MS-DOS records the parameter and the string you have set it to in the area of memory reserved for environment strings. If the parameter is already recorded, the new string replaces the one currently recorded. If you set the paramenter equal to blank, the parameter is removed from the list of environment strings.

If you use the SET command on its own without a command tail, MS- DOS lists all the environment strings that have been set - both through SET commands and automatically, for example through the use of other commands.

- **To set or reset a parameter**

Form SET *parameter=string*

- **To remove a paramenter from the environment**

Form SET *parameter* =

- **To list the current environment strings**

Form SET

Notes Details of which environment parameters a program needs to have set and what these should be set to should be given in the program's own user guide.

The list of current environment parameters and strings will contain a number that you will not know have been set. These are likely to be used by MS-DOS and should not be removed from the list unless you are fully aware of the role they play.

Example

- *Suppose a program behaves differently depending on whether the environment string represented by the parameter DISPLAY is CGA or EGA. To set DISPLAY to CGA, type the command line:*

```
SET DISPLAY=CGA
```

Later, you can set it to EGA by typing:

```
SET DISPLAY=EGA
```

- *To list on the screen, the present set of environment parameters and strings, type the command line:*

```
SET
```

A list similar to the following will appear on the screen:

```
A>set
PATH
COMSPEC=C:\COMMAND.COM
DISPLAY=CGA
```

Note the inclusion of parameters which were not set using the SET command. These have been set by other MS-DOS commands or by the operating system itself.

SHARE

External command

SHARE [/f:*space*][/L:*locks*]

Install file sharing and locking

You can set 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

Form SHARE [/f:*space*][/L:*locks*]

Options: The SHARE Switches are:

/f:space	Allocates file space (in bytes) where MS-DOS stores recorded file-sharing information.

The default value for the /f switch is 2048.

Since an average pathname is 20 bytes in length, each open file requires enough space for the length of the full filename, plus 11 bytes.

/L:locks	Allocates the number of locks you want to allow. The default for the /L switch is 20.

Example

To load the SHARE command type:

```
SHARE
```

File sharing will be loaded with default values for the /f and /L switches.

SHIFT

Subcommand

SHIFT

Shift the dummy parameters

Shift is used to change the value of each dummy parameter in a Batch file. This lets you work through a list of Batch file parameters.

Each time a SHIFT command is executed, the value of the %0 parameter is replaced by that of %1, the value of %1 is replaced by %2 and so on. The new value for %9 is taken from the command line used to execute the Batch file. The old meanings are forgotten.

This process lets you vary more than ten items within a Batch file (though, you can still use only ten dummy parameters at a time). It also allows you to use one Batch file to process a number of files, when this number changes from one time you use the Batch file, to the next. There is an example of how to do this, below.

Form SHIFT

Notes The old values for the parameters are lost.

Examples

- *To shift the meanings of the dummy parameters so that %0 takes the current value of %1, %1 takes the current value of %2, etc. include at the appropriate point in the Batch file the following command line:*

```
SHIFT
```

- *To shift the meanings of the dummy parameters so that %0 takes the current value of %2, %1 takes the current value of %3 , etc., either put at this point in the Batch file:*

```
SHIFT
SHIFT
```

or use a FOR command, such as:

```
FOR %%A IN (1 2) DO SHIFT
```

- *You want the Batch file MYBAT.BAT to carry out the same sequence of commands on a number of files, one after another. The problem is that the number of files you want to process varies each time you want to use the Batch file.*

  A possible solution is to have just the commands as the contents of the batch file, and to run it with each of the files in turn, ie. to enter the following command lines one after another. For example:

```
MYBAT File1
MYBAT File2
MYBAT File3
```

  A less tedious option is to give the file the following structure:

```
:LOOP
IF "%1"=="" GOTO END
...

commands

...
SHIFT
GOTO LOOP
:END
```

  The command line to run this would be:

```
MYBAT File1 File2 File3
```

  The first time through the commands %1 would be `File1`*, the second time it would be* `File2` *and the third time it would be* `File3`*. At the start of the next loop %1 would be blank and so the IF command will direct processing to the label* `:END`

SORT

External command and External filter

SORT *[/R][/+n]<[d:][\][path\]filename.filetype[>destination]*

Sort data

SORT is used to sort data into alphanumeric order. It can:

- **sort data stored in an existing file**
- **sort standard output from a different program**
- **sort into reverse order as well as into the normal order**
- **sort lines into alphabetical order based on the charcter at a given column in each line, not necessarily Column 1**

The style of the sort depends on which of the /R and /+*n* options are used. If no option is selected, the data is sorted into normal alphabetical order based on the character in Column 1 of each line.

The result of the sort can be sent to a file or to a specified output device. If no destination is specified, the sorted data is sent to the screen.

- **To sort data in a file**

Form SORT *[/R][+n],[d:][\][path\]filename/filetype[>destination]]*

- **To sort data generated by a program**

Form *command* | SORT *[/R][/+n][>destination]*

Options

/R	Reverse the sort, ie. sort from 9 to A
/+*n*	Sort at Column n

Examples

- *To sort the Root directory on Drive C before it is displayed, use the command line:*

```
DIR C:¦SORT
```

(assuming that the external command SORT is stored either in the default directory or in a directory that MS-DOS automatically searches - see Section 7.2.4)

- *To sort the lines in a text file called ROUGH.TXT into alphabetical order and store the result in a file called SORTED.TXT, use the command line:*

```
SORT <ROUGH.TXT>SORTED.TXT
```

- *To sort the lines in the file ROUGH.TXT in such a way that the characters which are eight character positions in on each line, are in alphabetical order, and to display the result on the screen, use the command line:*

```
SORT/+8 <ROUGH.TXT
```

SUBST

External command

`SUBST` *d:[d:][\]path*

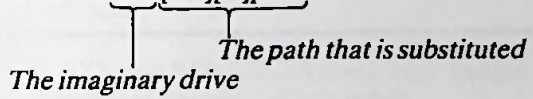
Substitute the name of an imaginary drive for a path

The SUBST command creates an imaginary drive by associating a path with a drive letter that is not already used by MS-DOS. This tells MS-DOS to substitute the path wherever you use the drive letter in describing the task you want carried out.

If you use the command on its own without a command tail, SUBST displays a list of all the substitutions that are currently in force.

- **To set up the substitution**

Form `SUBST` *d:[d:][\]path*



- **To end the substitution**

Form `SUBST` *d:* `/D`

The imaginary drive

- **To display current substitution**

Form `SUBST`

Notes You can form but cannot break a substitution that involves the directory holding the external command SUBST. You may need to reset your PC in order to break the connection with the imaginary drive.

If you try to use, in a new substitution, an imaginary drive that is already substituting for a directory, without first breaking the current association, you will see the message:

```
Invalid parameter
```

Break the current substitution first and then try again if you want use this drive for a different directory.

To use imaginary drives above E:, include the LASTDRIVE command (form: `LASTDRIVE = d:`) in the CONFIG.SYS file. For example, to be able to substitute drives up to N:, use the command line:

```
LASTDRIVE = N
```

Example

To assign the directory BUSINESS REPORTS on Drive A to the inmaginary drive, Drive E, type the command line:

```
SUBST E: A:\BUSINESS\REPORTS
```

(assuming that the external command SUBST is stored either in the default directory or in a directory that MS-DOS automatically searches - see Section 7.2.)

SYS

External command

SYS *d*:

Copy system files to the disk in the specified drive

The SYS command copies the system files on the disk in the default drive to the disk in a specified drive. It is therefore used:

- **to update the version of MS-DOS used on a particular startup disk**
- **to put the system files on a disk that has been formatted to allow room for these files (see the FORMAT command)**

The files copied are IO.SYS and MSDOS.SYS. COMMAND.COM is not transferred. Use a COPY command to add this file if you want to use the MS-DOS commands.

Form SYS *d*:

Notes IO.SYS and MSDOS.SYS are both hidden files and so are not listed when the directory of a Startup disk is listed.

Example

To copy the IO.SYS and MSDOS.SYS files onto the disk in Drive B, use the command line:

```
SYS B:
```

(assuming that the external command SYS is either in the default directory or in a directory that is automatically searched by MS- DOS - see Section 7.2.4).

TIME

Internal command

`TIME [hh:mm]`

Set or read the time

Time is used either to display or set your PC's clock. Its settings are used to time stamp files.

Any time that you type in, is checked before it is accepted. If the time is invalid, you will be asked to type in a new time.

Form `TIME`
or `TIME hh:mm`

where *hh* represents the hour (00..23) and *mm* the minute (00..59).

Notes If you type TIME, the current setting is displayed and then you are asked to type in a new time. If the time is correct, just press [↵].

The clock is set when you press [↵]. To make the setting accurate, type in a time 15..30 seconds in the future and then press [↵] when this time is reached. The Speaking Clock can help you set the time accurately: press [↵] on the third stroke!

Example

To set the time to be 4.23pm (16.23h), either use the command line:

```
TIME 16.23
```

which does the whole job in one go (press [↵] when it is exactly 4.23pm), or just:

```
TIME
```

If you use the second command, you will see something like this on the screen:

```
A>time
Current time is 12:24:31.33
Enter new time:
```

Type:

```
16:23
```

and press [↵] when it is exactly 4.23pm.

TREE

External command

`TREE [d:][/F]`

Display the pattern of directories on a drive

The TREE command lists the full path to each directory on the disk together with the names of any subdirectories this directory has. This information enables you to plot out the directory 'tree' for the disk.

If the `/F` option is used, the names of the files in each directory are also listed.

- **To display just the details of the directories**

Form `TREE [d:]`

- **To display filenames as well**

Form `TREE [d:]/F`

Notes If you do not specify a drive, the command lists information about the disk in the default drive.

If the only directory on the disk is the Root directory, MS-DOS will display the message:

```
No subdirectories exist
```

Examples

- *To display details of all the directories on the default drive, use the command line:*

```
TREE
```

(assuming that the external command TREE is stored either in the default directory or in a directory on the search path - see Section 7.2.4).

The result could be something like this:

```
A>tree
DIRECTORY PATH LISTING

Path:A:\SPREAD
Sub-directories: RECORDS

Path:A SPREAD RECORDS
Sub-directories:None
```

If there is only a Root directory on the disk, TREE will display the message:

```
No subdirectories exist
```

- *To display details of all the files as well as the directories on Drive C, use the command line:*

```
TREE C:/F
```

The result could be something like this:

```
A>tree c:/f
DIRECTORY PATH LISTING

Path:C\DIR1
Sub-directories: SUBDIR

Files           :MYFILE.1
                 MYFILE.2
                 MYFILE.3


Path:C:\DIR1\SUBDIR
Subdirectories: None

Files           : SUBFILE.1
                  SUBFILE.2
```

TYPE

Internal command

TYPE *[d:][\][path\]filename.filetype*

List a simple text file

TYPE displays the contents of the named file, on your screen. The only files suitable for being typed are simple ASCII text files.

If the file contains control characters, the display may be unreadable.

You can interrupt the listing by pressing Ctrl-S. Pressing Ctrl- Q restarts it. Pressing Ctrl-C abandons the listing.

You can only display one file at a time. Wildcards may not be used in the file specification.

Form TYPE *[d:][\][path\]filename.filetype*

Note Tab characters are replaced by spaces, putting the next character at the beginning of the next column. The columns are eight characters wide.

Examples

- *To display the AUTOEXEC.BAT file in the default directory on the screen, use the command line:*

```
TYPE AUTOEXEC.BAT
```

The result of this command could be something like this:

```
PATH \;
KEYB UK 437
ECHO OFF
DATETEST
IF ERRORLEVEL 1 GOTO SIGNON
DATE
TIME
:SIGNON
CLS
VER

A>
```

- *If the AUTOEXEC.BAT file you want to display is in the Root directory on Drive C, the command line you need is:*

```
TYPE C:\AUTOEXEC.BAT
```

If you TYPE anything other than a simple text file on the screen, you will see lots of strange characters as TYPE tries to interpret special codes within the file as characters. For example, you might see something like this:

```
Your PC will not be able to run programs or will if understan
                                                        your instruct
ions or without a disk operating system, MS-DOS is the
                                                    disk operating system supp
lied with your PC.
```

VER

Internal command

VER

Display MS-DOS version number

VER displays the version number of the MS-DOS operating system you are using.

Form VER

Example

To see which version of MS-DOS you are currently using, type:

```
VER
```

A message similar to the following will appear on the screen:

```
MS-DOS Version 3.30
```

VERIFY

Internal command

VERIFY *ON|OFF*

Turn on and turn off verification of all writes to disk

The VERIFY command is used to tell MS-DOS to verify that each file written to a disk can be read back successfully, by checking for example, that it does not contain any bad sectors. It is also used to turn off the verification procedure.

If the VERIFY command is used on its own, without any parameters, your PC will display whether verification is currently turned of or off.

- **To turn verification on**

Form VERIFY ON

- **To turn verification off**

Form VERIFY OFF

- **To show current state**

Form VERIFY

Notes While verification is turned on, you will only get a message on the screen when a disk error has been spotted.

Do not assume that verification is turned on because you have not entered a VERIFY OFF command. Verification can be turned off from within a program.

VOL

Internal command

VOL *d:*

Display disk volume label

VOL displays a DOS disk's label, ie. the name by which the disk can readily be identified and which you set by using the LABEL command (see Section 4.2)

Form VOL *[d:]*

Note The disk label is also displayed when you list any directory on the disk with a DIR command.

Example

To display the disk label on the default drive, use the command line:

```
VOL
```

To display the disk label of the disk in Drive B, use the command line:

```
VOL B:
```

XCOPY

External command

`XCOPY` *[d:][\][path\]source[d:][\][path\]destination[option[option...]]*

Copy files and directories

XCOPY is used:

- **To make copies of individual source files**
- **To copy the directory structure as well**

The source and the destination for each copy operation are directories on disk drives. The directories that can be copied are those which branch off the source directory. Empty directories can be copied as well as those containing files or further directories.

XCOPY sets an exit code to report the success of the copy. This can be used in an IF command (see Section 7.4).

Form `XCOPY` *[d:][\][path\]filename.filetype[d:][\][path\][destination-filename.filetype][option[option...]]*

where *filename.filetype* is a template using wildcards to specify a group of files.

Options

`/A`	Only copy files that have the Archive attribute set (see Section 6.3) **Note:** the Archive attribute is not reset after copying (see the `/M` option)
`/D:`*dd-mm-yy*	Only copy files that have been changed on or after the given date
`/E`	Copy empty subdirectories as well as ones with files in them. **Note:** The `/S` option must be selected as well
`/M`	Only copy files that have the Archive attribute set (see Section 5.3) **Note:** the Archive attribute is reset after copying (see the `/A` option)
`/P`	Confirm each copy
`/S`	Copy files from both the given directory and from its subdirectories (retaining the subdirectory structure)
`/V`	Verify each copy as it is written, to check that the new file is identical to the original
`/W`	Wait until a key is pressed before starting the copy (allowing you to swap the disks after issuing the command line)

Notes You must include at least one of the source drive, directory or file name template, but you can leave out all details of the destination: the current directory on the default drive is assusmed as the destination directory. *.* is the default file name template for both the source and the destination.

If you do not select the /S option, XCOPY will only copy files in the given directory. It will ignore any subdirectories to the directory and the files these contain.

If your chosen destination directory does not exist, XCOPY will create it for you. The directory must, of course, be a valid directory to create (see Section 6.1).

Additional Information

Exit codes

Ø	No errors
1	No files found to copy
2	Copy terminated by user (pressing Ctrl-C)
4	Either you specified an invalid directory, or there was an error in the command line syntax, or the file you specified wasn't found, or the path you quoted was not found.
5	Copy terminated by user after a disk error.

Examples

- *To copy all the files from the DIR1 directory (a subdirectory of the Root directory) on the default drive to the DIR2 directory on Drive C, you might use the command line:*

```
XCOPY \DIR1\*.* C:\DIR2
```

(assuming that the external command XCOPY is stored either in the default directory or in a directory that MS-DOS automatically searches - see Section 7.2.4).

- *To copy all the files and all the subdirectories with files in them from the DIR1 directory on the default drive to the DIR2 directory on a different disk, use the command line:*

```
XCOPY \DIR1\*.* B:\DIR2\/S
```

If you have a single-drive PC, MS-DOS will prompt you when to insert the disk for Drive A (your source disk) and when to insert the disk for Drive B (your destination disk). If you have a two-drive system, just use Drive B (the righthand disk drive) for the destination disk.

If you wanted to copy empty subdirectories as well as some with files in them, your command line would become:

```
XCOPY \DIR1\*.* B:\DIR2\/S/E
```

- *If you have a single-drive PC and the external command XCOPY is on a different disk to the files you want to copy, use the /W option. This makes XCOPY wait until you press a key before starting the copy. So with the MS-DOS kcommand disk in the drive, type the command line:*

```
XCOPY \DIR1\*.* B:\DIR2\/S/W
```

Then when the message:

```
Press any key when ready to start copying files
```

appears, release the MS-DOS disk from the drive and replace it with your source disk. Then press any character key (for example, the Space bar).

---

9. SUMMARY OF MS-DOS COMMANDS

Full details of each command are given in Chapter 8.

APPEND External command (Section 7.2)

Set a search path for data files

`APPEND` *[d:]\path[;[d:]\path...]*

ASSIGN External command (Section 7.1)

Assigns drive letter to another drive

`ASSIGN` *requested-drive=searched-drive*

ATTRIB External command (Section 6.3)

Set file attributes

`ATTRIB` *[+R|-R][+A|-A][d:][\][path\]filename.filetype*

BACKUP External command (Part 1, Chapter 3).

Backup files from one disk to another.

`BACKUP` *drive1:[path][filename[drive2:[/S][/M][/A] [/F]*
[/d:date] [/t:time] [/L:[[drive:][path]filename]][

BATCH Internal commands.

Run a batch of commands

[d:][path]filename[value[value...]]

BREAK Internal command (Section 7.2)

Sets how often MS-DOS checks for Ctrl-Break

`BREAK ON|OFF`

CALL Batchfile subcommand (Section 7.4)

Call a batchfile from within a batchfile

`CALL` *[d:][path]batchfile*

CHCP Internal command. (Part 1,Chapter 4).

Display or change the current code page for the command processor

`CHCP` *[nnn]*

CHDIR Internal command (Section 7.1)

Change the current directory

`CHDIR` *[d:][\]path*
`CD` *[d:][\]path*

CHKDSK External command (Section 5.3)

Check disk for errors and file for non-contiguous storage areas

`CHKDSK` *d:[\][path\][filename.filetype][\F][\V]*

CLS Internal command (Section 7.1)

Clear the screen

`CLS`

COMMAND External command.

Start the command processor

`COMMAND` *[d:][path][ctty-dev][/e:nnnn][/p][/c string]*

COMP External command (Section 6.2)

Compare files

`COMP` *[d:][path]filename.filetype[d:][path][filename.filetype]*

COPY Internal command (Section 6.2)

Copy files

`COPY` *[d:][\][path\]source[/A][/B][d:][\][path\]destination[/A][/B][/V]*

CTTY Internal command (Section 4.1)

Change the device used to issue commands

`CTTY` *device*

DATE Internal command (Section 4.3)

Set or read the date

`DATE` *[dd-mm-yy]*

DEBUG Debugging utility (not covered in this manual)

`DEBUG` *[d:][\][path\][filename.filetype][argument[argument...]]*

DEL Internal command (Section 6.2)

Delete a file

`DEL` *[d:][\][path\]filename.filetype*

DIR Internal command (Section 7.1)

Display directory

`DIR` *[d:][\][path\][filename.filetype]*`[/P][/W]`

DISKCOMP External command (Section 5.4)

Compare contents of target drive against contents of source drive

`DISKCOMP` *source-drive:* *target-drive:*`[/1][/8]`

DISKCOPY External command (Section 5.2)

Copy contents of source drive to target drive

DISKCOPY *source-drive:* *target-drive:*

ECHO Batch subcommand (Section 7.4)

Control echoing of commands on the screen

`ECHO` *[*`ON|OFF` *remark]*

EDLIN Text editor (Section 6.2)

Edit text

`EDLIN` *[d:][\][path\]filename.filetype[/B]*

ERASE Internal command (Section 6.2)

Delete a file

`ERASE` *[d:][[\][path\]filename.filetype*

EXE2BIN External command (not covered in this manual)

Convert EXE files to binary format

`EXE2BIN` *[d:][\][path\]filename*`.EXE`*[d:][\][path\][filename*.filetype]

where *filetype* is `COM` or `BIN`

EXIT Internal command (not covered in this manual)

Exit from COMMAND.COM amd return to the previous level (if any)

`EXIT`

FASTOPEN External command

Reduce time taken to open frequently-used files and directories

`FASTOPEN` *d:[=nnn]...*

FC External command

Compare files

`FC` *[option[option...]][d:][path]filename.filetype[d:][path]filename.filetype*

FDISK External command (Part1, Chapter7)

Configure a hard disk

`FDISK`

FIND External command and external filter (Part 1, Section 7.3

Look for a string in a file

`FIND` *[/V][/C][/N]*`"string"`*[d:][\][path\]filename.filetype*

FOR Batch subcommand (Section 7.4)

FOR is used to execute a number of similar commands

`FOR` *%%parameter* `IN (`*value[value...]*`) DO` *command*

FORMAT External command (Section 5.1)

Format the disk in the specified drive

`FORMAT` *d:[option[option...]]*

GOTO Batch subcommand (Section 7.4)

Go to a labelled point in the file

`GOTO` *label*

GRAFTABL External command

Load Graphics Table

`GRAFTABL [437|850|860|863|865| /status|?]`

GRAPHICS External command (Section 7.2) (Only for use in CGA mode)

Prepare your PC to print graphics screen displays

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

IF Batch subcommand (Section 7.4)

`IF [NOT]` *condition command*

JOIN External command (Section 7.1)

Join a disk drive into another disk's directory structure

`JOIN` *d: [d:][\]path*

KEYB External command

Load a keyboard program

`KEYB` *[xx[,[yyy],[[d:][path]filename[.ext]]]]*

LABEL External command (Section 4.2)

Create or change a disk label

`LABEL` *[d:]label*

LINK Object Linker (not covered in this manual)

`LINK`

MKDIR Internal command (Section 6.1)

Make a new directory

`MKDIR` *[d:][\]path*
`MD` *[d:][\]path*

MODE External command (Section 4.1)

Sets mode of operation of input and output devices

`MODE` *device:setting[,setting...]*

MORE External filter (Section 7.3)

Make output be displayed one screenful at a time

`MORE`

MOUSE External command (Part 1)

Start the mouse working

`MOUSE` *[/S][/F][/Lxxxx][/Rxxxx][/Xxx][/Yxx]*

NLSFUNC External command

Load country-specific information

`NLSFUNC` *[[d:][path]filename[.ext]]*

PARK External command (Part 1, Chapter 7).

Prepares your Hard Disk (if fitted) before moving it

`PARK` *[d:]*

PATH Internal command (Section 7.2)

Define a search path

`PATH` *[d:]\path[;[d:]\path...]*

PAUSE Batch subcommand (Section 7.4)

Pause while processing

`PAUSE` *[remark]*

PRINT External command (Section 6.2)

Print text files in the background

`PRINT` *[d:][\][path\]filename.filetype[[d:][\][path\]filename.filetype...]*

PROMPT Internal command (Section 4.2)

Set up a new system prompt

`PROMPT` *prompt-text*

RECOVER External command

Recover a file or disk containing bad sectors

`RECOVER` *d:[\][path\][filename.filetype]*

RESTORE External command.

Restores files made by a previous BACKUP command

`RESTORE` *drive1*`:`*[drive2*`:`*][path]*`:`*filename[*`/S`*][*`/P`*] [*`/B`*:date][*`/A`*:date][*`/E`*:time]-[*`/L`*:time][*`/M`*][*`/N`*]*

REM Batch subcommand (Section 7.4)

Display remark

`REM` *remark*

RENAME Internal command (Section 6.2)

Rename a file

`REN` *[d:][\][path\old-name new-name*

REPLACE External command (Section 6.3)

Insert new files and update old versions

`REPLACE` *[d:][\][path\]filename.filetype [d:][\][path][option[option...]]*

RMDIR Internal command (Section 6.1)

Remove directory

`RMDIR` *[d:][\]path*
`RD` *[d:][\]path*

SELECT External command

Install MS-DOS on a new floppy disk or hard disk with desired country-specific information and keyboard layout.

`SELECT` *[[drive1:][drive2:][path]][xxx][yy]*

SET Internal command (Section 7.1)

Set a parameter to a string

`SET` *parameter=string*

SHARE External command

Install file sharing and locking

`SHARE` *[/F:space][/L:locks]*

SHIFT Batch subcommand (Section 7.4)

Shift the dummy parameters

`SHIFT`

SORT External command and filter (Section 7.3)

Sort data

`SORT` *[/R][/+n]<[d:][\][path\]filename.filetype[>destination]*

SUBST External command (Section 7.1)

Substitute the name of an imaginary drive for a path

`SUBST` *d:* *[d:][\]path*

SYS External command (Section 5.1)

Copy system files to the disk in the specified drive

`SYS` *d:*

TIME Internal command (Section 4.3)

Set or read the time

`TIME` *[hh:mm]*

TREE External command (Section 6.1)

Display the pattern of directories on a drive

`TREE` *[d:][/F]*

TYPE Internal command (Section 6.2)

List a simple text file

`TYPE` *[d:][\][path\]filename.filetype*

VER Internal command (Section 7.1)

Display MS-DOS version number

`VER`

VERIFY Internal command (Section 7.1)

Turn on and turn off verification of all writes to disk

`VERIFY ON|OFF`

VOL Internal command (Section 6.2)

Display disk volume label

`VOL` *d:*

XCOPY External command (Section 6.2)

Copy files and directories

`XCOPY` *[d:][\][path\]source[d:][\][path\destination[option[option...]]*

10. TROUBLESHOOTING

This Chapter looks at what to do and what might have gone wrong when your PC or your program does not work in the way you expect. If you cannot find the solution to your problem here, consult your dealer.

Note: Your PC will normally bleep if it fails to read a disk or if it cannot accept the character you just typed. If you do not hear any bleep, adjust the volume control.

10.1 Trouble during Startup or when resetting your PC

- **If nothing happens at all**

  Check that the mains socket is working by plugging in and switching on a lamp you know is working.

  Check that the PC's mains plug is correctly wired and that the fuse in the plug has not blown.

  If neither of these actions shows you where the problem is, consult your dealer.

- **The PC bleeps and asks you to insert a System disk when you insert your Startu disk**

  Check that you inserted the right disk and then reset the machine. If it blee again, the disk it is trying to read is either damaged, or it does not have a operating system software on it. Try another Startup disk: if you do not have another Startup disk or if your PC fails to read this disk as well, consult your dealer. The problem could simply be that your disk drive needs cleaning, in which case your dealer will be able to tell you which cleaning product is recommended for use on your PC.

  As soon as possible after this failure, use MS-DOS's CHKDSK command(see Chapter 8) to find out whether the disk that failed was damaged.

- **Software is read into your PC's memory but the machine dies**

  Check that you inserted the right disk and then reset the machine. If you PC dies again, the software on your disk has been corrupted.. Try another Startup disk: if you do not have another Startup disk, consult your dealer.

- **When all the software has been read in, the screen display is plausible but is not what you expected**

  You have used either a disk that has been set up to run an application program immediately after loading the system software or a disk with different system software on it. Which has happened, will be apparent from the screen display.

  Either leave the application program by holding down the **[Ctrl]** key and pressing the **[Break]** key, or reset you PC using another Startup disk.

10.2 When commands fail

Commands fail because:

- you have made a typing error in the command line
- you have not specified the location of a file correctly
- you have not got enough memory space for the program to run
- you have not got enough room on the disk to store the new files the program produces
- the file holding the program has been corrupted
- you are trying to use an external command or a program under the wrong operating system

You can usually tell why a command has failed from what appears on the screen.

- **If the operating system puts up a name followed by "Bad command or file name" or a question mark, it has not found the command, program or batch file you wanted to run**

  The commonest reason for this is that you mistyped the command name. For example, you might have typed COYP instead of COPY. The other possibility is that the program file is not in the directory or directories the operating system searched. You may have misdirected the operating system, for example because you forgot which disk you had in the drive. Check where you told the operating system to look for the file.

- **If the operating system puts up a message like "File not found" together with a file specification, it has not found one of the files you asked it to process**

  Either you mistyped the filename or the filetype. Or the file is not in the directory you specified. Check where you told the operating system to look for the file.

  In either of these cases you can either retype the command line at the new system prompt or copy and then edit your previous command line. The keystrokes to help you do this are described in Section 7.2.1. When you have finished editing the command line, press the [↵] key.

- **If the operating system puts up some other message like "Memory insufficient to run program" or "Disk full", it has met some other problem in carrying out your command**

  Look up the message in Sections 10.7 - 10.8 and then take whatever action is appropriate. You may, for example, need to erase some files you no longer need from the disk or change what you are trying to do.

- **If the screen goes blank or your PC generally seems dead**

  If this happens immediately after you type the command line to run the program, check in the program's user guide precisely which operating system you are supposed to run this program under. This includes checking the version number. It could be that the program will only work with a specific version of, say, MS-DOS - for example MS-DOS 2.0.

  If the screen goes blank after the program has run for a little while, first try typing Ctrl-Q (by holding down the **[Ctrl]**-key and pressing **Q**). It could be that you accidentally caused the program to pause.

If these actions do not help you identify the problem, reset your PC (See Part I, How to Reset your PC) and then try again. If the same failure happens again and it is a program that you have written or edited, try again with your back-up copy of the program and or check the program. If the program is one you have bought, try again using your master copy of the program: if that fails too, consult your dealer.

10.3 Trouble with the keyboard

- **If your PC fails to respond to what you type**

  Check whether the cursor is flashing on the screen. If it is, check that the keyboard cable is connected firmly into the keyboard socket on the side of the PC System Unit. If your PC still does not respond, switch off, wait for a little while and then switch back on and go through the normal Startup procedure. If your PC still fails to respond to your keyboard, consult your dealer.

  If the cursor is not flashing, something in your program has caused your PC to "die". Put your Startup disk into Drive A and reset your machine by holding down **[Ctrl]** and **[Alt]** and pressing **[Del]** . If your PC still fails to respond, release the disk(s) from the drive(s), switch your machine off and then switch it on again and go through the standard PC Startup procedure (see Part 1).

  If your PC still fails to respond to your keyboard, consult your dealer.

- **If you get cursor movement when you thought you were using the Numeric keypad (and vice versa)**

  Press the **[Num Lock]** key and then try again.

- **If you get upper case letters when you were expecting lower case letters**

  Press the **[Caps Lock]** key and try again.

- **If the characters you type in, overwrite what you have already typed in, rather than get inserted into your text**

  Press the **[Ins]** key and try again.

- **If the keys do not produce the characters you expect**

  You may have used the KEYBxx utility from a different DOS- compatible operating system. You must use the operating systems supplied with your PC to get full keyboard and mouse support.

10.4 Trouble with a disk drive

Floppy Disk drive

If your PC frequently reports that it cannot read or write to a disk drive, it could be that your disk drive is the problem rather than your disks. A likely cause of trouble is that the disk drive is in need of cleaning.

A number of cleaners are available but not all are recommended for use on your PC. Your dealer will be able to advise you on which one to use.

If cleaning the disk drive does not solve the problem, either get the drive fixed or replace it with a new one as soon as possible. Your dealer will be able to help you here. Do not limp on with a defective drive - you could easily damage your disks -and do not try using a two-drive machine as a single-drive system disconnecting the defective drive. If you do, (i) you will invalidate the warranty on your PC, and (ii) you will find that the initial system check will fail and you will not be able to use the machine anyway.

There is no alternative to getting a defective disk drive fixed.

Hard Disk drive

If your system is not responding as you think it should, then you might have a problem with the Hard Disk. However, it is possible that the problem lies elsewhere:

If the computer seems to be ignoring the Hard Disk, claiming that it is an "Invalid drive" - you may not have carried out the installation procedure in Part I of this guide. Remember - until you have, the Hard Disk is NOT available for use.

Getting one of the following messages means that you have a hardware fault of some description, and you should consult your dealer.

Hard Disk errors:

```
Fatal disk error

Disk reset failed

Disk diagnostics failed

Disk not ready
or: Hard Disk drive not ready

Disk recalibration failed
or: Hard Disk recalibration failure

Write sector buffer failed

Hard Disk controller failure

Error: Hard Disk I/O Error
```

Note: If you attempt to park a third party Hard Disk (ie. one not supplied as originally fitted equipment) you may receive the following message:

```
Invalid drive specification
```

If a third party drive is fitted to your PC, the error "message" ** 1701 displayed immediately after you have switched on your PC simply means: Hard Disk not working. You will have to consult your dealer for a more detailed diagnosis.

10.5 Trouble with a printer

The following covers the problems most commonly experienced with printers. Generally the place to look for advice is the printer's own manual.

- **You send data to the printer but it does not respond**

  Check that the printer is on and on-line. Note: You often have to abandon the current printing and start again, if you switch a printer on-line after your PC has started sending data to the printer.

- **You send data to the printer and the printer head moves but nothing is printed**

  Check that you have a ribbon in the printer.

- **You use the MS-DOS MODE command to change the number of characters per line (the pitch) or the number of lines per inch (the line pitch) but no change is seen in what the printer produces**

  Check the printer's manuaal to see whether the printer can print different numbers of characters per line or lines per inch. The DMP 1, for example, only has one possible line pitch.

- **You set up a document on your screen containing pound signs but when you print it, all the pound signs have been replaced, e.g. by hashes**

  Your computer and your printer are not using the exactly same character set. Look in the printer's manual to see if there is a setting on the printer that will give you pound signs (for example, the European IBM character set option on the DMP3000).

10.6 Trouble with the mouse

- **Moving the mouse has no effect on the pointer on the screen**

  Check that the mouse cable is connected firmly to the mouse socket on the side of the PC System Unit.

  If this does not restore the mouse, reset your PC by holding down the **[Ctrl]** and **[Alt]** keys and pressing the **[Del]** key. If the mouse still does not work, consult your dealer.

  If you are using a DOS program, you will be able to continue working by using the cursor keys. If you are running the program under MS-DOS, you might also check whether you have loaded the mouse driver MOUSE.COM (see Part I using the Mouse).

- **Clicking the mouse buttons does not have the expected effect**

  Check that the mouse cable is connected firmly to the mouse socket on the side of the PC System Unit.

  If this does not restore the mouse, reset your PC by holding down the **[Ctrl]** and **[Alt]** keys and pressing the **[Del]** key. If the mouse buttons still do not work, consult your dealer. In the short term, you should be able to continue working by pressing keys on the keyboard.

  If you are running the program under MS-DOS, you might also check whether you have loaded the mouse driver MOUSE.COM (See Part I, Using the Mouse with Commercial Programs).

Note: Only use the MOUSE.COM supplied with your PC. This has been tailored specifically for your PC.

10.7 Disk and Device errors

Whenever a disk or device error occurs, the operating system displays a message of the form:

error `READING|WRITING DRIVE` *n*`:`
`Abort, Retry, Ignore?`

error specifies the type of error that has occured: the options are:

```
Bad call format error
Bad command error
Bad media type
Bad unit error
Data error
FCB unavailable
General failure
Invalid disk change
I/O error
Lock violation
Non-DOS disk error
No paper error
Not ready error
Read fault error
Sector not found error
Seek error
Sharing violation
Write fault error
Write protect error
```

10.8 System messages

Most system messages are self-explanatory but what to do when others appear may not always be clear. If you are unsure, look for the error (or something very similar) in the following list and study the explanation given here.

- `Active Code Page not available from con device`

The code page that the system is currently using is not supported on the console (screen) you are using.

- `Add filename? (Y\N)`

The REPLACE command displays this prompt if you specify the `\W` switch. Type `Y` (for Yes), if you want to add the file to the disk, or `N` (for No), if you do not want to add the file.

- `All files cancelled by operator`

  MS-DOS displays this message when you specify the /E switch with the PRINT command.

- `Allocation error, size adjusted`

  The size of a file, as indicated in the directory, was wrong. The file was cut short to match the amount of data allocated.

- `All specified file(s) are contiguous`

  All files are written sequentially on the disk. To correct this error automatically, specify the CHKDSK/f switch.

- `Bad or missing keyboard definition file`

  MS-DOS cannot find the KEYBxx file that you specified with the KEYB command.

  Check that the file you specified does exist on the disk. Also check whether your path includes the directory in which this file is to be found. Then re-type the command. If you get this message again, the KEYBOARD.SYS or KEYB.COM file may be corrupted.

- `Bad Partition Table`

  There is no DOS partition on the hard disk. You must use the FDISK command to create a DOS partition on your hard disk.

- `Cannot CHDIR to path - tree past this point not processed`

  CHKDSK is checking the structure of the directory and is unable to go to the specified directory. The subdirectories of this directory will not be verified.

  Use the CHKDSK `/F` to correct this error automatically.

- `Cannot CHDIR to root`

  CHKDSK is checking the tree structure of the directory and is unable to return to the root directory. CHKSDK is unable to continue checking the remaining subdirectories. Try to restart MS-DOS. If this error continues to occur, the disk is unusable.

- `Cannot COPY from (or to) a reserved device`

  You cannot copy files from or to, a device.

- `Cannot create a zero cylinder partition`

  You are trying to create a partition size of 0 cylinders.

  You must allocate a minimum of 1 cylinder to any partition you create.

- `Cannot create Subdirectory BACKUP on drive x:`

  This may mean the disk is full, write-protected or that the backup subdirectory already exists and is read-only.

  Use another disk as a target disk.

- `Cannot perform a cyclic copy`

  When you are using the XCOPY with the /S switch, you cannot specify a subdirectory of the source as the target.

- `Cannot recover ... entry`
  `Entry has a bad attribute (or link or size)`

  The parent directory is defective and cannot be recovered.

  If you have specified the /f switch, CHKDSK tries to correct the error automatically.

- `Cannot use FASTOPEN for drive x:`

  FASTOPEN works only with local, fixed disks and can work with a maximum of four disks at a time.

  It is not possible to use FASTOPEN over a network, with a floppy disk, or with more than four disks at one time.

- `xxxxxxxx code page drive cannot be initialised`

  MS-DOS cannot start either the PRINTER.SYS or DISPLAY.SYS program.

  Check the device command line in your CONFIG.SYS file. You probably included an illegal parameter.

- **Code page not prepared**

  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, the device command line in your CONFIG.SYS file should allow for additional subfonts.

- **Code page xxx not prepared for all devices**

  The code page you have selected is not currently supported by a device. If the device supports code page switching (and is currently on line) 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 is unable to select a code page for the system.

  Make sure that NLSFUNC is installed. Now retry the CHCP command, if you have not used the DEVICE command in your CONFIG.SYS file to install device drivers. 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 retry the CHCP command.

- **Code page operation not supported on this device**

  You have specified a device and code page combination which MS.DOS does not recognize as valid.

  Check that the device you specified exists and that you have listed a valid code page. Also check that the code page is supported on the specified device.

- **Code page requested xxx is not valid for given keyboard code**

  The keyboard code and code page specified are not compatible.

  Retry the KEYB command with a keyboard code and code page that are compatible.

- **Code page specified has not been designated**

  You have typed the KEYB command with an option the system doesn't recognize. You must prepare the associated code page for your console screen device. Prepare the associated code page for CON using the MODE PREPARE command. Then retype the KEYB command.

- **Code page specified has not been prepared**

  You have typed the KEYB command with an option the system doesn't recognize. You must prepare the associated code page for your console screen device. Prepare the associated code page for CON using the **MODE PREPARE** command. Then retype the KEYB command.

- **Code page specified is inconsistent with invoked code page**

  The KEYB option you have selected does not coincide with the code page for your console screen device (CON).

  Use the **MODE SELECT** command to change the code page for CON.

- **Code page xxx**

  The code page currently being used by the device specified.

- **Code pages cannot be prepared**

  You have tried to prepare more than the total number of code pages supported for this device or specified a duplicate code page for this device.

  Check the DEVICE command line in your CONFIG.SYS file for the total number of code pages allowed for this device. Use the **/status** option of the **MODE** command to find which code pages are already prepared for this device.

- **Compare process ended**

  Message displayed when a fatal error occurs during the comparison.

- **Comparing t tracks**
  sectors per track, s side(s)

  This message confirms the format of the disks being compared.

- **Contains *n* non-contiguous blocks**

  The disk contains fragmented files.

- **Copy process ended**

  To copy this disk, use the **COPY** or **XCOPY** command instead of the **DISKCOPY** command. The new copy will store the files sequentially.

- **Current keyboard does not support this code page**

  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 using the KEYB command.

- **Data error reading drive x:**

  MS.DOS could not read data from a disk properly. This is often due to a defective disk.

  Try typing R (for Retry) several times, or if the message still appears type A (for Abort) to end the program. (It is a good idea to make a new copy of the disk, because if it is defective, you may lose information).

- **xxxxxxxx device driver cannot be initialized**

  You are trying to install a device driver by using a DEVICE command line in your CONFIG.SYS file where syntax of that command line is wrong.

- **Device Error during Status**

  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 that your device supports code page switching, if it does 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.

- **Device Error during Prepare**

  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 that your device supports code page switching, if it does 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 correct.

- **Device Error during Select**

  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 that your device supports code page switching, if it does 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 correct.

- `Device Error during write of font file to device`

  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 that your device supports code page switching, if it does 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 correct.

- `Device or code page missing from font file`

  MS-DOS did not find a definition in the font file of the indicated code page for this device.

  Use the **MODE** command to specify a different code page for this device and check to see that the font file supports the code page you want to use. This error may cause specified code pages to be undefined, use the **MODE** command to prepare and refresh lost code pages.

- `DEVICE Support Not Present`

  MS-DOS device control does not support the disk drive.

- `Directory not empty`

  You can only join onto an empty directory.

- ```
  Diskette bad or incompatible
  MS-DOS cannot copy this disk.
  ```

  The disk you are copying is not formatted or is a format which is not compatible with MS-DOS.

- ```
  Do not specify filename(s)
  Command format: DISKCOMP d: d:[/1][/8]
  ```

  You specified an incorrect option.

- **Do not specify filename(s)**
  Command format: DISKCOPY d: d:[/1]

  You specified an incorrect option.

- **Do you see the leftmost 0? (Y/N)**

  The MODE command 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) to shift the test pattern to the right.

- **Do you see the rightmost 9? (Y/N)**

  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 test pattern to the left.

- **Does *name* specify a file name or directory name on the target (F = file D = directory)?**

  XCOPY displays this message if the target directory does not exist.

  Type F to specify a file, or D to create a new directory.

- **(.)(..) Does not exist**

  When using CHKDSK, this message indicates that either the "." or ".." directory entry is invalid.

- **DOS 2.0 or later required**

  You cannot use these utilities with versions of MS.DOS earlier than 2.0.

- **Drive types or diskette types not compatible**

  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.

  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 to copy the files across, or reformat the target disk so that it's the same format as the source disk.

- **End of input file**

  The entire file was read into memory. If the file was read in sections, this message indicates that the final section of the file is in memory.

- `Enter current Volume Label for drive x:`

  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.

- `Error during read of Font file`

  MS-DOS found an error when it tried to read the font file for the code page specified.

- `Error in country command`

  The syntax for the COUNTRY command in your CONFIG.SYS file is incorrect

- `Error opening log file`

  MS-DOS cannot open the BACKUP LOG file.

  Check the drive and path specified. Remember that the log file cannot be located on the target drive. If you did not specify a filename for the log, then the error occurred when MS-DOS tried to open and create BACKUP LOG on the source disk.

- `Error reading/writing partition table`

  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 found errors on the disk. If you have not specified the option CHKDSK continues printing messages but will not correct the errors.

  To correct the problems encountered by the CHKDSK command you should use the /f option.

- `Errors on list device indicate that it may be off-line. Please check it`

  Your printer is not switched on-line.

- `Error trying to open BACKUP log file`
  `Continuing without making log entries`

  You specified the BACKUP/1 option, but BACKUP could not create the BACKUP LOG file.

- `Failure to access code page font file`

  MS.DOS cannot open the font file for the specified code page.

  Check to see that you typed font file, and 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.

- `FCB unavailable reading (or writing) drive x:`

  An unusual error has occurred, usually requiring an experienced programmer to fix it.

  Type R (for Retry) or A (for Abort).

- `fc: incompatible switches`

  You have specified options that are not compatible. (For example, /b amd /L.)

  You should not combine binary and ASCII comparison options.

- `File allocation table bad drive x:`

  This message means that the disk was not formatted or was not formatted properly. It could also mean that there is another operating system on the disk other than MS.DOS.

  Check the disk by running the CHKDSK /f command. If this message is displayed again, you must reformat the disk.

- `File filename cancelled by operator`

  MS.DOS displays this message when you specify the /t switch with the Print command.

- `File cannot be converted`

  The input file is not in the correct format..

- `File is READ-ONLY`

  You cannot change a file designated read-only.

- `Files cannot be added to this diskette unless the PACK (/P) switch is used Set the switch (Y/N)?`

  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).

- `FIND: Access denied`

  You cannot access the file.

  Check that the disk is not write-protected, read-only, or locked.

- `FIND: Read error in filename`

  The Find command could not read the specified file.

- `FIND: Syntax error`

  Check to make sure that you have typed the command correctly.

- `First cluster number is invalid, entry truncated`

  The file directory entry contains an invalid pointer to the data area. If you specified the /f switch, the file is cut short to a zero-length file.

- `Fixups needed - base segment hex:`

  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`

  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.

- `General failure reading (or writing) drive x:`

  An unusual error has occurred. This error usually requires an experienced programmer to fix it.

  Type R (for retry) or A (for Abort)

- `Graftabl needs DOS version 2.Ø or later`

  You cannot use GRAFTABL with earlier versions of MS.DOS than version 2.0.

- `Graphics characters already loaded`

  The GRAFTABL command displays this message if you have already loaded the table of graphics characters into memory.

- `Graphics characters loaded`

  The GRAFTABL command displays this message after it loads the table of graphics characters into memory.

- `Hardware code pages:`
  `Prepared code pages:`

  This message lists the current code pages prepared for the specified device.

- `Has invalid cluster, file truncated`

  The file directory entry contains an invalid pointer to the data area. If you specified the /f switch, the file is cut short to a zero-length file.

- `Incorrect APPEND Version`

  You are using incompatible version of MS-DOS.

- `Incorrect parameter`

  One of the command options you specified is wrong.

- `Infinite retry on parallel printer timeout`

  Your printer is probably off-line or not ready.

  If the printer appears to be ready, you may have to reset the computer by holding down the **[Ctrl]** and **[Alt]** keys and pressing the **[Del]** key.

- `Insert destination disk in drive x:`
  `and strike any key when ready`

  This message appears when you are using SYS to transfer the operating system with a single disk drive.

  You should insert a disk in the appropriate drive and press any character or number key to begin processing.

- `Internal error`

  This message indicates an error in the utility.

- `Internal stack overflow`
  `System Halted`

  The system tried to use more stacks than were available. This caused a series of hardware interrupts and halted the system. Restart MS-DOS. Then edit your CONFIG.SYS file and allocate more stack resources.

- `Invalid argument`

  You have specified an invalid argument.

  Refer to Chapter 8, "MS-DOS Commands," for the correct syntax of the command, and try again.

- `Invalid baud rate specified`

  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 country code or code page`

  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`

  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 device`

  The device specified was not AUX, CON, NUL, or PRN.

- `Invalid device parameters from device driver`

  This message indicates the number of hidden sectors cannot be divided evenly by the number of sectors per track. This means that the partition does not start on a track boundary. Check the CONFIG.SYS file for incorrect DEVICE or DRIVPARM commands.

- `Invalid drive specification`

  The drive is incorrect or does not exist
  Enter a valid drive name.

- `Invalid signature in COUNTRY.SYS file`

  The SELECT command cannot find a proper file header or a specific country code in the COUNTRY.SYS file.

- `Invalid signature in KEYBOARD.SYS file`

  The SELECT command cannot find the proper file header or a specific or a specific keyboard code.

- `Invalid STACK parameter`

  There is an invalid parameter in the syntax of the Stack command in your CONFIG.SYS file.

- `Invalid syntax`

  You have made a mistake in typing the command. Check the form of the command in the list of commands in Chapter 8.

- `Lock violation reading (or writing) drive x:`

  A program tried to access part of a file that another program was using.

  Type A (for Abort) or R (for Retry)

- `x lost cluster(s) found in y chains`
  `Convert lost chains to files (Y/N)?`

  CHKDSK displays this information if it finds information on the disk that is not allocated properly in the disk's File Allocation Table.

- If you answer Y (yes) 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. The filename extension will be .CHK. If you did not specify the /F switch, CHKDSK displays the message "x bytes would be freed".
- If you type N (no) CHKDSK frees the lost blocks so that they can be reallocated and does not recover any data lost in those blocks. If you did not specify the /F switch, CHKDSK will do nothing.

● `Logging to file x`

The BACKUP command is writing a BACKUP LOG to the specified file.

● `Missing from the file is either the device ID or the code page`

Either, the specified code page is not supported in the code page information file (.cpi file) or, the .cpi file does not support the specified printer. See the MODE command (Chapter 8) for a valid list of cplist values

● `Must specify COM1, COM2, COM3 or COM4`

You have to specify a serial port.

● `Name of list device [PRN]`

When the PRINT command is first run, and the /d switch is not specified, you are prompted to specify the name of any valid device. Press the [↵] key to accept the default list device PRN.

● `Non-DOS disk error reading (or writing) drive x:`

MS-DOS does not recognise the disk format because either there is information missing from the disk, or it contains another operating system.
Try using the CHKDSK command to correct the problem. If this does not work, reformat the disk by using the FORMAT command. **NOTE:** formatting the disk will destroy all the files on the disk.

● `Non-standard version of Graphic Character Set Table is already loaded`

MS-DOS does not recognise the current table of graphics characters because it was modified after it was loaded.

● `No path`

You typed PATH and pressed [↵] to find out what your search path is; but you must specify a command search path.

- `***Not able to back up ( or restore) file***`

  There may be an error in the source file or on the target disk.

  The CHKDSK command may be able to identify the problem.

- `Not ready error reading (or writing) drive x:`

  The device specified in the message is not ready to accept or transmit data

  This usually means the drive door is open: close the door and type R; or check that the printer is ready to print.

- `One or more CON code pages invalid for given language`

  Your keyboard and screen console device are working from different code pages.

- `Read error, COUNTRY.SYS`

  MS-DOS cannot read the COUNTRY.SYS file

  Re-run the command. If the same message appears, the COUNTRY.SYS file is probably corrupted, so restore the file from backup.

- `Resident part of PRINT installed`

  The PRINT command was processed using other processes and the available memory has been reduced by several thousand bytes.

- `Restore file sequence error`

  You have restored files in the wrong order; backup disks must be inserted in the same order that they were backed up.

- `Sector not found error reading (or writing) drive x:`

  The disk probably has a defective spot; MS-DOS cannot find the requested information on it.

  Copy all the files on the disk to a good disk and then try to reformat the defective disk.

- `Sharing violation reading drive x:`

  A program tried to access a file while another program was using that file.

  Type A (abort) or wait a while and type R (retry)

- `Specified MS-DOS search directory bad`

  The SHELL command in the CONFIG.SYS file is incorrect.

  Check whether the COMMAND.COM file exists and that MS-DOS can find it.

- `System transferred`

  The system files were transferred during FORMAT or SYS command processing.

- `The only bootable partition on drive 1 is already marked active`

  You are trying to change the active partition. The active partition must reside on the first hard disk drive on your system.

- `Too many drive entries`

  You can use FASTOPEN with up to four hard drives; you are trying to specify a fifth hard drive.

- `Too many name entries`

  The maximum number of entries for a drive is 999; the total number of entries specified exceeds that number.

- `Unable to create KEYB table in resident memory`

  MS-DOS cannot create a country-specific table for the specified keyboard code.

  Check that there is enough available memory to create this table.

- `Unrecoverable error in directory`
  `Convert directory to file (Y/N)?`

  CHKDSK cannot correct an error in a directory.

  - If you answer Y (yes) to this question, CHKDSK will convert the bad directory into a file, which you can then correct or delete.
  - If you answer N (no), you may not be able to write to, or read from the bad directory.

- `Unrecoverable read (or write ) error on drive x:`

  MS-DOS cannot read or write data to the specified device.
  Check that the disk is properly inserted into the disk drive.
  Type R (retry). If the same error occurs Type A (abort).

- ```
  Warning! Diskette is out of sequence
  Replace diskette or continue if okay
  Strike any key when ready
  ```

  You should restore diskettes in the same order that you backed them up.

- `Warning! Read error in EXE file`

  This is to inform you that the amount read, was less than the size of the header.

- `Write fault error writing drive x:`

  MS-DOS cannot write data to the device you have specified; check that the disk is properly inserted into the disk drive and then type R (retry). If the second attempt is not successful, type A (abort).

- `Write protect error writing drive X:`

  The disk which you are trying to write data to, is a write- protected disk.
  ```

INDEX

Pt.1 stands for Part 1
Pt.2 stands for Part 2

A

B

C

D

E

F

G

H

I

J

K

L

M

N

O

P

R

S

T

V

W

X


5 013413 600076