

904 878 8564
SoftKlone
327 Office Plaza Drive
Tallahassee, FL 32301

Main BBS:

904 878 9884 8 data, no parity, 1 stop
2400 o/c

XMODEM

(CRC)



AMSTRAD plc

MIRROR II v 3.6.12

AMSTRAD PPC USER MANUAL

(USA)

AMSTRAD

1800 999 0174

© Manual Copyright 1988, AMSTRAD plc

Neither the whole nor any part of the information contained herein, nor the product described in this manual may be adapted or reproduced in any material form except with the prior written approval of AMSTRAD plc('AMSTRAD').

The product described in this manual and products for use with it are subject to continuous development and improvement. In particular, there may be differences between the messages displayed on your screen and those shown here.

All information of a technical nature and particulars of the product and its use (including the information and particulars in this manual) are given by AMSTRAD in good faith. However, it is acknowledged that there may be errors or omissions in this manual.

AMSTRAD welcomes comments and suggestions relating to the product or this manual.

All correspondence should be addressed to: **AMSTRAD INC**
AMSTRAD National Service Center
1915 HARRISON ROAD
LONGVIEW TX 75604

All maintenance and service on the product must be carried out by AMSTRAD authorised dealers. AMSTRAD cannot accept any liability whatsoever for any loss or damage caused by service or maintenance by unauthorised personnel. This manual is intended only to assist the reader in the use of the product, and therefore AMSTRAD shall not be liable for any loss or damage whatsoever arising from the use of any information or particulars in, or any error or omission in, this manual or any incorrect use of the product.

Manual revised for the PPC By J Jones including material written by J Gilmour of Locomotive Software Ltd for AMSTRAD plc.

Published by AMSTRAD

First Edition 1988

Trademark Acknowledgements

MS-DOS® is a trademark of Microsoft® Corporation.

CROSSTALK is a trademark of Digital Communications Associates Inc.

MIRROR II AND MIRROR are trademarks of Softklone Distributing Corporation.

Hayes is a trademark of Hayes Microcomputer Products Inc.

IBM is a registered trademark of International Business Machines Corp.

Compuserve is a trademark of Compuserve Information Service.

AMSTRAD PPC, PPC640, PPC512, PC1512, PC1640 are trademarks of AMSTRAD plc.

AMSTRAD name & logos are registered trademarks of AMSTRAD plc. All Rights reserved
Unauthorised use of the trademark or word AMSTRAD is strictly forbidden.

IMPORTANT

This equipment generates and uses radio frequency energy, and if not installed and used properly, that is, in strict accordance with the manufacturer's instructions, may cause interference to radio and television reception. It has been type tested and found to comply with the limits for a Class B computing device in accordance with the specifications in Subject J of Part 15 of FCC Rules, which are designed to provide reasonable protection against such interference in a residential installation. However, there is no guarantee that the interference will not occur in a particular installation. If this equipment does cause interference to radio or television reception, which can be determined by turning the equipment off and on, the user is encouraged to try to correct the interference by one or more of the following measures:

...re-orient the receiving antenna.

...re-locate the computer with respect to the receiver.

...move the computer away from the receiver

...plug the computer into a different outlet so that the computer and receiver are on different branch circuits.

If necessary, the user should consult the dealer or an experienced radio/television technician for additional suggestions. The user may find the following booklet prepared by the Federal Communications Commission helpful:

'How to Identify and resolve Radio/TV Interference Problems'

This booklet is available from the US Government Printing Office, Washington DC20402
-Stock No. 004-000-00345-4

IMPORTANT - WHEN CONNECTING ANY PERIPHERAL TO THIS EQUIPMENT, A CORRECT UL/FCC APPROVED SHIELDED CABLE MUST BE USED.

WARNING - This equipment has been certified to comply with the limits for a Class B computing device, pursuant to Subpart J of Part 15 of FCC Rules. Only peripherals (computer input/output devices, terminals, printers, etc.) certified to comply with the Class B limits may be attached to this computer. Operation with non-certified peripherals is likely to result in interference to radio and TV reception.

INTRODUCTION

The AMSTRAD PPC is a compact computer, which is easily transportable between any number of locations, for use in a variety of ways. For example, it could help you to prepare letters or to carry out routine office tasks at work; then at home it could help you budget ahead and so keep in credit at the bank. It can also run computer games when you want to relax.

The internal battery pack enables you to use your AMSTRAD PPC anywhere, But, to avoid needless use of battery power, AMSTRAD have designed the PPC to accept power input from a number of alternative external sources.

THE PPC MAY BE CONNECTED TO ONE OF THE FOLLOWING THREE POWER SOURCES ONLY.

Where A.C. power is available:

- a. **The external A.C. adaptor supplied with your PPC 640 (U.K.-Model PPC640 AC-B; U.S.A.-Model PPC640 AC-A; Europe - Model PPC640 AC-E).**
- b. **An AMSTRAD PC-MD, PC-CD or PC-ECD monitor, if available.**

Where A.C. power is not available:

- c. **From a 12v car cigar lighter socket, via the car battery adaptor supplied with your PPC. (Model PPC640DC).**

NO OTHER EXTERNAL POWER SOURCE MAY BE USED.

Important Note:

Always use 'C' size ALKALINE cells (Duracell or similar type) for your internal power supply. Most other types cannot continuously deliver the current required by the PPC. Rechargeable cells have a lower working voltage, which will give early indication of power failure, so their use is not recommended. Try to organise your work to avoid using the disk drives (and modem, on PPC 640 models) any more than you can help, when using the internal supply, thus minimising power consumption. It is recommended that the PPC640 modem is NOT operated on battery power.

The PPC 640 models have a built-in modem, designed to allow the PPC 640 to be plugged directly into a standard telephone socket. It is supplied complete with MIRROR II, a powerful, but simple to use, communications program. This gives you the power to transfer files between your office micro and your current location, to access databases such as Compuserve, and to use Bulletin Boards and Electronic Mail.

AMSTRAD'S built in LIQUID CRYSTAL DISPLAY (LCD) monitor gives a sharp, clear display, making your system completely self-contained. However, switches are provided to allow you to configure your PPC for a choice between internal LCD, or external colour or monochrome VDU. The PPC will accept the AMSTRAD PC-MD, PC-CD or PC-ECD, IBM, or IBM-PC compatible monitors.

Choice of colour or monochrome adaptor, to suit an external monitor if used, is made by dials, as is the choice of whether the initial screen output is displayed on the LCD or an external monitor. Output can then be switched simply from internal to external display, or vice-versa, by software switching controlled by the **[Scroll Lock]** and right hand **[Ctrl]** keys.

The AMSTRAD PPC has been designed so that it can run the same programs as many of today's microcomputers. For instance, it can run the same programs as an IBM desktop PC. You could, for example, take the 3 1/2 inch disk you use to run your word processing program on an IBM PC and use it directly in your AMSTRAD PPC. With the PPC 640, if your office micro is connected to the public telephone service, you can access your files from anywhere that there is a public telephone line, with a standard plug-in connection. If you have a sales force, you can, by suitably programming your office system, allow them to use the PPC to obtain information and enter order details from a hotel or a customer's office, giving a swift and professional service.

The MIRROR communications package, supplied with the PPC 640, allows you to access files on any system connected to the public telephone service, provided that you are entitled to use that system. Communication with public services and IBM compatible micros, connected to standard modems, should present little difficulty. Mainframe systems and non IBM compatible micros may present technical problems and require full familiarity with MIRROR II's wide range of facilities.

Most Systems are protected from unauthorised access by **PASSWORDS** and if you are planning to access your office system remotely, whether using the AMSTRAD PPC 640 or any other system, you should ensure that it is similarly protected.

When MIRROR II is running on the 'host' system, that is, the system holding the files to which a remote user requires access, it is possible to allocate different levels of access to different remote users. For example, you might choose to be able to write to any of your office files yourself, from any location, whilst allowing some of your staff to have read-only access and allowing others to have write access, but perhaps only to certain files. The Amstrad PPC 640, on-line with MIRROR, makes it easy for authorised users to gain access to the host system at their allocated levels, providing assistance in MIRROR or other compatible communications software.

Also supplied to you with the AMSTRAD PPC is Microsoft's MS-DOS Version 3.3

MS-DOS's role is to supervise the work your AMSTRAD PPC carries out. It also provides tools for keeping your stored information and programs organised.

About this manual

This manual covers the use of the AMSTRAD PPC range, comprising PPC 512S, PPC 512D, PPC640S AND PPC640D models. The PPC 640 models have an integral intelligent modem and are supplied with the MIRROR II communications software package.

The manual is in 3 sections:

- * Section 1: Fundamental aspects of using the AMSTRAD PPC
 - * Section 2: Using MS-DOS Commands
 - * Section 3: Communicating, with MIRROR II
-

Each section starts with its own contents list, describing the subsections it contains. Cross references are given in the form: section 1.2.3, referring to section 1, sub-section 2, sub-sub section 3.

Section 1 describes how to connect up the various parts of your AMSTRAD PPC and then takes you through your first steps in using your system. It also tells you about the disks and the programs you should buy for your AMSTRAD PPC and how to use these. This part of the manual includes an introduction to computing for anyone using a microcomputer for the first time.

Section 2 describes how to carry out a broader range of tasks by typing instructions to MS-DOS.

Section 3 explains how to connect the AMSTRAD PPC 640 to the telephone network and how you can use the system to transfer data to and from remote locations.

Appendices at the end of the manual give information about the modem specifications for the PPC 640 models, and pointers to more advanced use of your system.

If you want to use your PPC simply to run programs you buy, then all the information you need to get these running is given in section 1 of the manual. The programs' own user guides should tell you everything else you need to know.

Whilst you would normally run versions of programs specifically intended for the AMSTRAD PPC, it will generally be possible to run programs intended for the IBM PC and for PC compatibles. Again all the information you will need will either be in section 1 of this manual or in the programs' own user guides. Programs may be loaded directly from 3 1/2 inch disks or transferred from another system, either by direct connection between serial ports or via modems and telephone lines, using MIRROR II or other communications software.

Note the text of this guide uses UK spellings

CONTENTS

Section 1: Fundamentals of using the AMSTRAD PPC

1.1 Fundamentals of computing

What is a computer?; How your disks are organised; Bringing your filing system up to date

1.2 Setting up the AMSTRAD PPC

Fitting the A.C. power plug; Fitting the batteries; Setting up the PPC

1.3 Preparing for future use

First steps; Setting the AMSTRAD PPC's internal clock; Backing-up your AMSTRAD PPC disks; Learning more about the system

1.4 About disks

1.5 What the AMSTRAD PPC can be used for

Running a standard DOS program; Running Communication programs on the PPC 640

1.6 Using MS-DOS to organise your disks

Copying a disk; Preparing a new disk for use; Copying a file; Deleting a file; Renaming a file; Finding out how large a file is; Finding out how much room there is on a disk; Starting to edit a file.

1.7 Fundamentals of using the AMSTRAD PPC

The Startup procedure; Switching off; Resetting the AMSTRAD PPC; About filenames; Using wildcards to specify a number of files; About folder and directory names; Specifying a path; Cleaning the AMSTRAD PPC

Section 2: Using MS-DOS Commands

2.1 Introducing MS-DOS

Starting to use MS-DOS; Using MS-DOS commands; More about - system prompt, command name, command tail; When commands fail

2.2 Overview of MS-DOS commands

Tasks to use MS-DOS commands for; Running programs; Organising your work; Tailoring the AMSTRAD PPC to your needs; Some practice in using MS-DOS commands

2.3 Conventions

2.4 Running programs

2.4.1 Running a program

2.4.2 Shortcuts at the keyboard

2.4.3 Redirecting input and output

2.4.4 Setting up a sequence of commands

2.5 Organising your work

2.5.1 Putting files into groups

2.5.2 Disk housekeeping

2.5.3 Protecting your files

2.6 Processing disks

2.6.1 Preparing new disks for use (Formatting)

2.6.2 Copying disks

2.6.3 Checking disks

2.6.4 Comparing disks

2.7 Tailoring the AMSTRAD PPC to your needs

2.7.1 Setting up Input and Output Devices

2.7.2 Personalising your PPC

2.7.3 Setting the AMSTRAD PPC's clock

2.8 MS-DOS command summary

Section 3: Communicating with MIRROR II

3.1 Getting started - a quick introduction

3.2 Notes for the Newcomer

3.3 General details of PC2000 modem

3.4 Introducing MIRROR II

3.5 NEWUSER procedure

3.6 Basic Calling Procedure

3.7 Working with MIRROR II

3.8 Alphabetical list of MIRROR commands

APPENDICES

Appendix A: Buying and Installing programs

Suitable programs; Selecting which system software to use; First steps; Making the software convenient to use

Appendix B: The Amstrad PPC software disks

Operating system and applications supplied with the PPC

Appendix C: The AMSTRAD PPC character sets

To find the key that produces a particular character; To find the internal code for a particular character

Appendix D: The AMSTRAD PPC hardware

The LCD; The keyboard; Device names

Appendix E: Expanding your AMSTRAD PPC

Installing a printer; Connecting an external monitor; Re-configuring system software

Appendix F: Reference material on the AMSTRAD PPC

Guides to the software; Guides to the hardware

Appendix G: Troubleshooting

Trouble during Startup or when resetting your PPC; When commands fail; Trouble with the keyboard; Trouble with a disk drive; Trouble with a printer; Disk and Device errors; System messages; Hardware error messages

Appendix H: The PC 2000 modem

Legal requirements; Commands; Registers; Result codes; Modes

INDEX

End User Licence Agreements

SECTION 1.

FUNDAMENTALS OF USING THE

AMSTRAD PPC

1.1 FUNDAMENTALS OF COMPUTING

This section is designed to introduce newcomers to the sort of tasks that can be carried out on a computer and to some of the jargon that you may read or hear. Although you are naturally keen to start using your AMSTRAD PPC, read this section carefully before you go on to section 2. Come back to this section later from time to time, especially if you have never used a microcomputer or if you haven't used a DOS operating system before, until you are confident that you understand these fundamental concepts.

What is a computer?

A computer is a complex system of electronics that is used to store, manipulate and retrieve data for you. This data can take a variety of forms. For example, it can be:

- numbers (if you are using the computer as a highly sophisticated calculator)
- text (if you are using the computer as a 'word processor')
- a mixture of the numbers and text

The data is manipulated through sequences of instructions known as programs. Each program carries out a specific task or series of tasks. It might, for example, maintain your telephone list or it might process your accounts for you. Programs that do such tasks are often described as Applications.

The electronics is called the computer's hardware; the programs are called its software.

To run your programs, your computer needs some special software known as the Operating System. This has the job of interpreting your instructions and getting the computer to carry out tasks for you. The operating system is often described as 'breathing life into the computer' - to remind you that it has to be loaded into the computer before any program can be run. The AMSTRAD PPC's operating system is called MS-DOS, often abbreviated to DOS.

To tell your AMSTRAD PPC what programs you want run or tasks carried out, you need a means of 'talking' to the operating system. It also needs to 'talk back' to you to tell you the result of its operations, report errors and tell you that it is ready for another instruction.

To achieve this, the computer has:

- a keyboard through which you can type in information and instructions
- a Display screen through which the operating system can send you messages and display results. Your PPC has an integral LCD display, which takes little power to operate, for convenience when you wish to transport the PPC and operate from battery power. Where a wall socket is available, you may wish to connect a standard PC monitor instead. Simply plug in, operate the switches provided, and you can use either a monochrome or colour monitor. Details are given in appendix E.

Note:

If you use an AMSTRAD monitor (PC-MD, PC-CD or PC-ECD are suitable) you **MUST** power the PPC from the monitor power supply, in order to ensure correct operation of the monitor.

The information and instructions supplied to a computer are called Input and the messages and information produced by the computer are called Output. Because you use the keyboard to supply information, the keyboard is described as an Input Device. Similarly, the Display screen is described as an Output Device because it is used to display information produced by the computer.

The AMSTRAD PPC also has connectors for a printer (an Output Device) and for a communications link (both an Input and an Output Device). You may hear Input and Output Devices being called Peripherals, just as you may hear the connectors being called Ports.

So that the operating system can interpret your instructions, there needs to be a set of rules about how your instructions should be given and standard ways in which your computer replies to you. These go together to form what is called:

the DOS Command Interface

which responds to commands typed in at the keyboard

While a program is running, both the data being processed and the necessary instructions are held in the computer's memory (which is known as its RAM). Some special programs, known as the operating system's Internal Commands, are held in the memory all the time you are running programs. All other programs have to be read into the computer's memory each time they are used and are erased from memory the moment the program has finished.

There is only a limited amount of storage space in a computer's RAM and any information held in RAM is lost the moment the computer is switched off.

A permanent way of storing programs and data for your AMSTRAD PPC is provided by 3 1/2 inch floppy disks. (You may also see these referred to as Diskettes or Discs.) You make the information on the disks available to your computer by inserting the disks in the disk drives and telling your PPC to read the information into its memory. You can also instruct the computer to record information stored in its memory on a disk - by 'writing the data to the disk'.

Each of these processes involves making a copy of the information: writing to the disk makes a copy of the information in the computer's memory on the disk and reading from a disk makes a copy in the computer's memory of the information stored on the disk. Information written on a disk stays on the disk until it is either overwritten or deliberately erased.

How your disks are organised

Programs and data are kept in files, both in the computer's memory and on disks. Each file typically contains either a program or a quantity of related data, and each file has a name to identify it by.

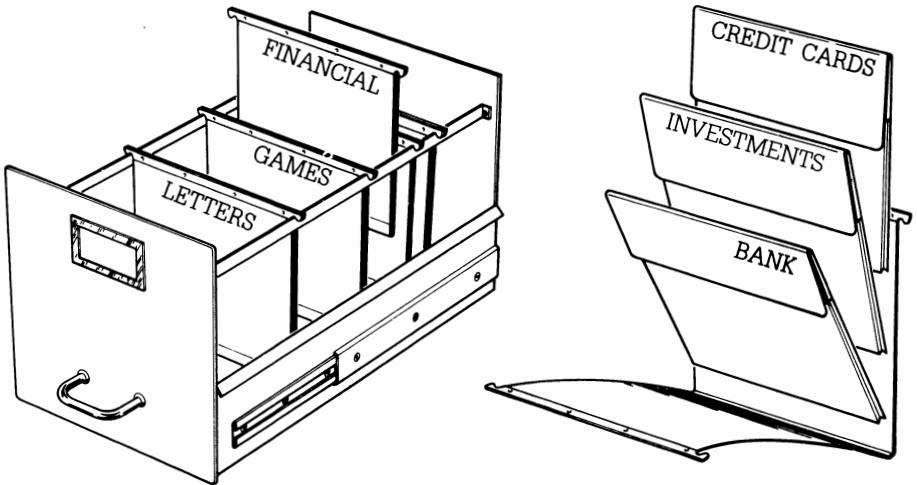
The way computer files are organised when they are stored on a disk is most easily understood by comparing your disks with a filing cabinet.

Each computer file is like an individual letter or a document in the filing cabinet. Each disk is like a drawer of the filing cabinet and putting a disk into a drive is like opening one drawer of the filing cabinet. The number of disk drives you have can be thought of as the number of drawers in the filing cabinet that you can have open at the same time.

Just as you keep the letters and documents in a filing cabinet in folders, so you keep computer files on a disk in folders. The folder on the computer disk also contains a list of the items that have been put in that folder. This list is called a directory and, for this reason, the folders are often called Directories.

You could keep everything in one folder both in the filing cabinet and on your computer disk but then you would have so many different items in the folder that it would become increasingly difficult to find the file or letter you want. To make the files more manageable, you group them into different folders.

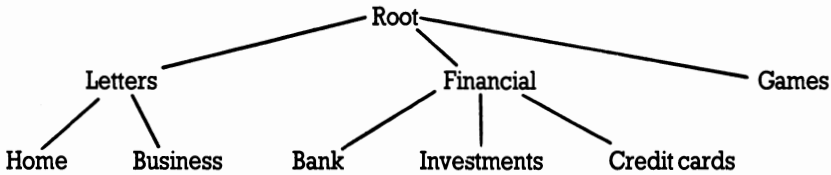
The first thing you see when you pull open a drawer or put the disk in the drive is a set of folders. The only clue you have to their contents is the name the folder has.



If you select a particular folder from your filing cabinet and open it, you will see either a number of documents or further folders (or a mixture of the two). Similarly, if you 'open' a folder on your computer disk, you will 'see' either a number of files or further folders. The further folders divide up the contents of the main folder once again into more manageable groups. Of course, you can't actually read what is on the disks by eye; as we will explain shortly, there are commands which cause the system to list the contents of any folder on the display for you.

To see what was in one of the new set of folders, you would once again select the folder and open it - whether you were handling the contents of the filing cabinet drawer or the contents of your disk.

The standard way of representing this pattern of folders is by a sort of family tree, starting at the top with the tree's Root and then branching out into the folders used to group the files on the disk. The folders in our filing cabinet would be represented as:



On a computer disk there would be a directory for each folder, just holding details of items on the next level down the tree. For example:

- the Root directory will just have the entries 'Letters', 'Financial' and 'Games'
- the 'Financial' directory will just have the entries 'Bank', 'Investments' and 'Credit cards' (plus the names of any files sitting alongside these three folders in the Financial folder)

If you think about it, the Root directory always summarises the entire contents of the disk.

Before we leave this section, a couple of definitions. Just as you generally select one folder from your filing cabinet to use at a time, so you select one main directory to work with at any one time. This directory is known as the Default Directory (because it is the directory your PPC assumes you want to work with unless you tell it otherwise). Similarly, you select one disk drive as the principal drive you are working with: this drive is known as the Default Drive. There is no problem if you have only one drive; when you are copying from one disk to another your system will tell you which disk to load and when.

Bringing your filing system up to date

Sorting out the files you have stored on disk is like sorting out a filing cabinet - something you do to ensure that you are keeping the information you want in the places you want and not wasting space by storing information you no longer need. The processes involved are often called Disk Housekeeping.

The operations you can carry out are:

- creating new folders (directories)
- deleting folders you no longer want
- creating new files
- editing existing files to bring them up to date
- deleting files you no longer want
- copying files
- renaming files

Creating a new folder is like putting a new folder into your filing cabinet drawer, ready to hold either new documents or documents you move to the new folder.

Deleting a folder is equivalent to throwing away an existing folder, contents and all. MS-DOS assumes that you would never do that without taking out the contents first, so you can't use the DOS command RMDIR to remove a folder, until you have deleted all the files in it.

Creating a file is just like producing a new letter or document and then storing it in the appropriate folder in the filing cabinet. Your accounting program, for example, may generate invoices which you would store in separate files in a single folder on your disk - just as you would store individual invoices you drew up by hand.

Editing a file produces an up to date version of a document and replaces the old document with this new version. Files are edited by running a special program called a text editor and using this to change your file.

Copying a file or a document is equivalent to taking a photocopy of an existing letter and storing this in a folder in another part of your filing system. Neither the original letter nor the original file is in any way changed by the copying process, and in both cases, you can either keep the same name for the copy or give it a new one. However, if you decide to keep the copy in the same folder as the original, you will need to give the copy a different name.

Renaming a file is equivalent to giving a letter a new reference number but leaving it in the same folder. However, files are often all you can rename: you typically can't rename a folder on a disk in the same way that you can relabel a folder in the filing cabinet, though you could create a new folder with the new name, using the MKDIR command, copy all the files into that, then delete the original files and folder.

Deleting a file is equivalent to taking a letter out of the filing cabinet and throwing it into the fire. The information that was stored in the file cannot be recovered.

MS-DOS commands are provided, to allow you to find out what is on any disk. Each command is entered by pressing [←] when you have typed it in correctly. Although we have shown some commands here, by way of illustration, it will be best if you work carefully through the setting up instructions for your PPC before trying them out.

The DIR command has several forms:

A > DIR will list the contents of the current default directory from the disk on the current default drive. (Drive A)

Typing / W at the end of any of the forms of DIR command will cause the directory contents to be printed in columns across the full screen width, rather than in a single long column.

If we have the disk just described on drive A, then:

A > DIR A : \ F I N A N C I A L will list the contents of the FINANCIAL directory, that is, Bank, Investments and Credit Cards directories, in our example.

Alternatively, you can use the CD command to change the current default directory:

A > CD \ F I N A N C I A L

Then the command

A > D I R will list only the contents of the **FINANCIAL** directory.

It is easy to get confused when learning these commands. Try to remember that:

A > C D ** will return to the root directory of the disk on drive **A.

Programs are stored on the disks like any other files, so to run the (imaginary) program **OFFICE.COM** stored in the directory **USEFUL**, you would first load the disk containing the **USEFUL** directory onto a drive, (having powered up the PPC and loaded **MS-DOS**) then type:

A > A : \USEFUL\OFFICE . COM

The program would then load and run.

Note:

The **A >** in the above examples represents the prompt written on the display by **MS-DOS**, it is not typed in by you. If you have a second floppy disk drive, it will be identified as drive **B**. To address another drive; at the prompt, type the new drive letter followed by a colon, for example:

B > A :

will change the default drive from drive **B** to drive **A** and display the prompt:

A >

BUT you can operate on a file on another drive without changing the default directory. If, for example, you wished to load the program file **OFFICE.COM** from directory **USEFUL** on a disk in drive **B**, when the default directory was the root directory on drive **A**, then the command line:

A > B : \USEFUL\OFFICE . COM

will load the program, but the default directory will still be the root directory on drive **A**. Study the **DOS** commands carefully and practise them on some test files which you create yourself (you can use the **RPED** file editor, provided on your **SYSTEM** disk by **AMSTRAD**, to do that) so that you do not have problems when you are handling valuable data and program files.

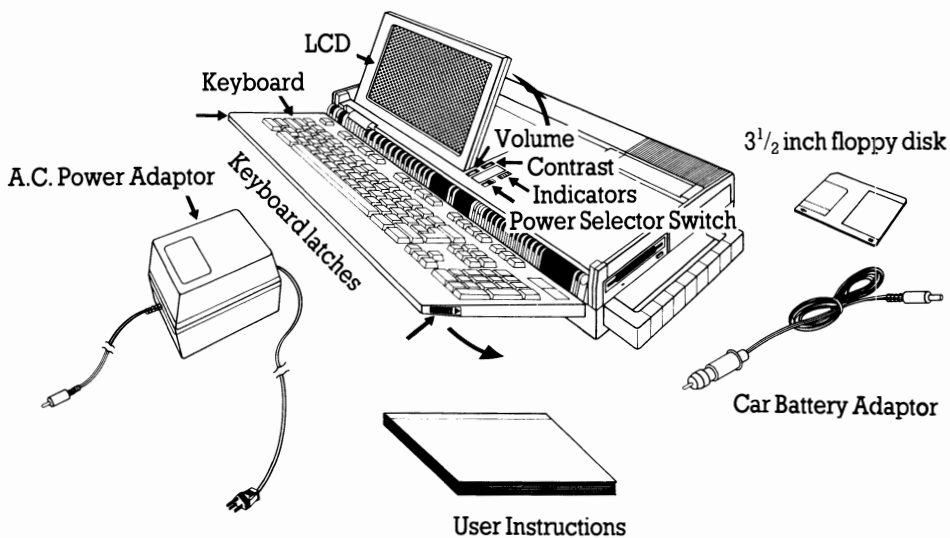
You will find full details of these and all the other **MS-DOS** commands in section 2 of this manual.

1.2 SETTING UP YOUR AMSTRAD PPC

This section describes how to set up your AMSTRAD PPC. However experienced you are in using microcomputers, you are advised to follow ALL the instructions given both in this section and in section 1.3, which describes how to prepare for your future use of the AMSTRAD PPC. If you rush on too quickly to running programs, you risk corrupting the software supplied with your AMSTRAD PPC; the consequence of this will be that you have to buy new copies of this software from your dealer.

Although it is small enough to be used almost anywhere, you should initially set up your AMSTRAD PPC in a clear, dust-free, area on a desk or table that is about 3ft wide and 2ft deep. You will then have room to put down your disks and open manuals, in comfort. If you wish to fit enhancements, such as an external monitor or a printer, you will need extra space. With a printer, you need to leave room for paper feeding too. If you are going to use the communications facilities incorporated in the PPC 640 models, you need to be close to a telephone socket also.

The box in which your AMSTRAD PPC is supplied should contain the following:



Before you do anything else, check that you have all these items. If any item is missing, consult your dealer.

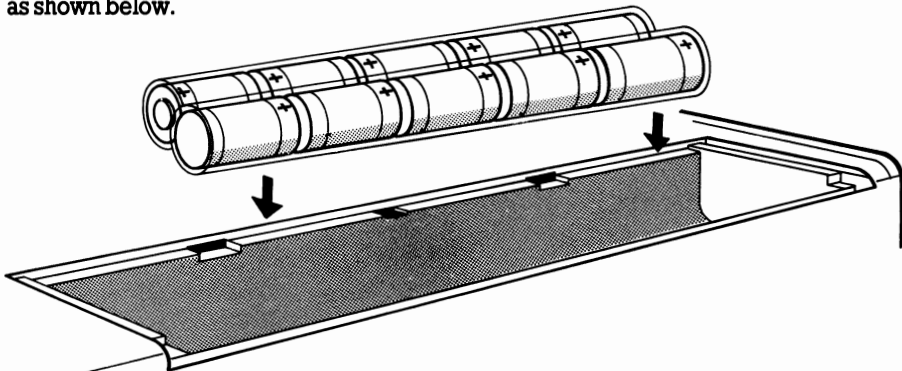
Follow these instructions until you have the basic system set up and are satisfied that it is working correctly. Then refer to appendix E if you wish to attach a printer or an external monitor unit.

For your PPC you will also need at least one new blank floppy disk. Ask your dealer for 3¹/₂ inch 720 kb disks. (These are the same disks as those used in an IBM Personal System 2). Buy disks that are labelled to show who produced them. Don't buy cheap unlabelled disks, they could corrupt your data and even damage the drives.

1.2.2 Inserting the batteries.

Your PPC takes ten C size batteries. Use alkaline cells, Duracell or similar. The cheaper zinc/carbon cells cannot continuously deliver sufficient power. Rechargeable (nickel/cadmium) cells have a lower voltage, even when fully charged, than the disposable cells. If used in the PPC they will give early indication of failure; we do not therefore recommend them for this application.

Fit five batteries into each of the battery tubes provided in the accessory box. Check that you put each battery the right way round, with its positive (+) end toward the plus sign on the battery housing. Insert the tubes in the battery compartment in the bottom of the PPC, as shown below.



Notes:

i) The batteries are used to power your PPC when no other supply is available; whilst travelling by train for example. The batteries have only a limited life, because the larger batteries needed to give longer service would be too heavy and bulky to transport in comfort. So use one of the alternative power sources whenever you can and conserve your batteries for when you really need them. Try to use the disks as little as possible, when running on the internal battery supply; you can minimise power consumption that way. It is recommended that the modem supplied in PPC 640 models is not operated on internal battery power.

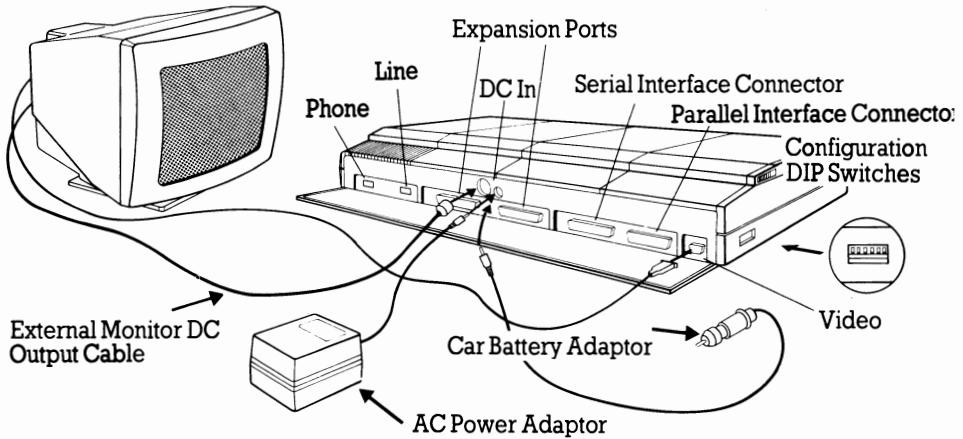
ii) If you close the keyboard with power still switched on, whether from an internal or external supply, an alarm will sound. A further alarm circuit warns when the internal batteries are becoming exhausted. An early indication will be given by the increased load as disks are accessed causing the alarm to sound. Don't leave old batteries in the PPC; they can leak and do damage. When fitting new batteries, remove the battery tubes from the PPC, discard the old batteries and retain the tubes for re-use. They reduce the risk of accidental power loss if the PPC is jolted when operating on internal batteries.

iii) Moving the power selector switch from EXT ON to EXT OFF and vice-versa causes the system to reset. You should not, therefore, switch power sources when you have files unsaved or disks loaded in the drives.

iv) Saving files to disk when the batteries are low may cause the LCD to fade. This condition is temporary and does not affect the transfer to disk. The display will reappear when the transfer is complete and the load on the batteries falls.

1.2.3 Setting up the PPC

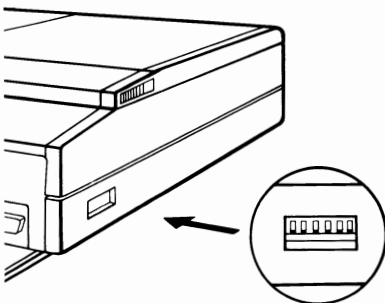
Place the PPC on your table and turn it so that the floppy disk drives are easily accessible. They are located above the handle, on the right hand side as you face the PPC. If you have two drives, the one nearest the keyboard is drive A.



Unlatch the keyboard and unfold it ready for use. Lift the Display so that its screen is at a suitable angle. (You can swing it up to any convenient angle, by lifting with the tab at the right rear of the screen).

IMPORTANT

Configuration switches are provided, to allow you to select from a number of display options. Refer to the following illustration and table to initially select the internal LCD display.



Normal dip switch settings (↑ = up, ↓ = down)

Switch No	1	2	3	4	5
MDA (Mono)	↓	↑	↓	↓	↓
CGA (Colour)	↓	↓	↓	↓	↑

Note:

The mode must be selected to suit the external monitor, when used. The LCD works in both CGA and MDA modes.

Configuration options

Switch No	Function	Selection
1	Select default monitor	↑ = external ↓ = Internal
2	Select graphics adaptor	↑ = MDA ↓ = CGA
3		always ↓

Switches 4 and 5 together select the default VDU mode, as follows.

4	5	Function
↓	↓	MDA
↓	↑	80 Col CGA
↑	↓	40 Col CGA
↑	↑	Reserved

Switch 6 is not used

If you are a newcomer to computing we would advise you not to attach any other devices at this stage, though when you are ready to do so you will find full instructions for attaching a printer and an external monitor in appendix E.

You will find it best to practise using the MS-DOS Operating System until you are reasonably familiar with it or, if you are anxious to use a printer straight away, get help from someone who has a lot of experience of MS-DOS

1.3 PREPARING FOR FUTURE USE

This section tells you how to prepare for future use of your PPC. However experienced you are in using microcomputers, you are advised to follow ALL the instructions given here; there are NO shortcuts. If you rush on too quickly to running programs, you risk corrupting the software supplied with your PPC: the consequence of this will be that you have to buy new copies of this software from your dealer.

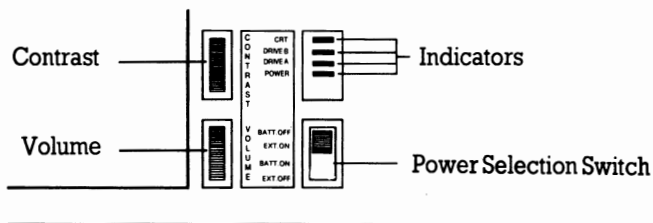
Note:

If, in following these instructions, things happen that aren't explained here or you see messages that are nothing like the ones described here, turn to appendix G 'Troubleshooting' and see if you can find out what has gone wrong. If you can't, consult your dealer.

1.3.1 Switching on a new system

Start with

- the adaptor A.C. power plug out of the supply socket
- the power switch, on the top of the PPC to the right of the LCD (visible after you have unfolded the keyboard) in its EXT ON position.
- no disk in any floppy disk drive.



Connect the jack plug from the A.C. adaptor to the DC IN jack socket on the back of the PPC. Insert the A.C. power plug into the supply and switch on the supply socket.

The message `P l e a s e w a i t . . .` should appear on the screen.

Your PPC then goes through a built-in system check. If all is well, it will beep and a message similar to the following will shortly appear on the screen.

```
Amstrad Portable PC 640K (V1.8) 09:00 on 10 January 1988
```

```
(c) 1987 AMSTRAD plc
```

```
Last used at 18:00 on 09 January 1988
```

```
Insert a SYSTEM disk into drive A
```

```
Then press any key
```

If the batteries are dead, the time and date messages will not appear. Instead, you will be reminded to set the time and date and fit new batteries. You are not obliged to do so; MS-DOS can be loaded and the PPC used, but the time and date used in any labels or messages will be incorrect. If the batteries are not replaced, the time and date will be correctly maintained, once set, until the PPC is switched off. You cannot set the time and date until you have loaded MS-DOS. Section 1.3.3 will tell you how to use the TIME and DATE commands.

If you do not hear a bleep or you feel the bleep is too loud, adjust the Volume Control. This is situated on the top of the unit, to the right of the LCD screen. Setting the Volume Control approximately in the middle of its range should suit most people. Adjust the contrast control to achieve the most easily readable screen.

If the message `Check keyboard` is displayed, check that nothing is resting on the keyboard.

Both the PPC 512 and the PPC 640 are available in single and double disk drive versions. There are some small variations in the Startup procedure for single or double drive models; be careful to follow the instructions relevant to your PPC in the following sections.

The `SYSTEM` disk that you get with your PPC contains `MS-DOS` and the programs needed to install the complete system. (For all disk drive configurations).

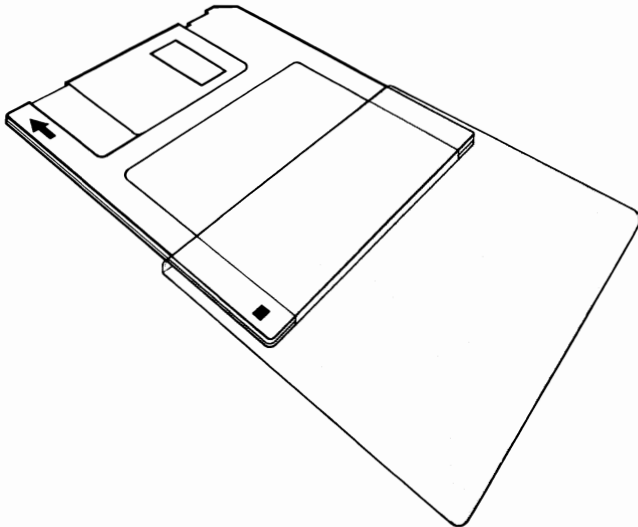
As soon as you have completed the Startup procedure you will be shown how to make a copy of your `SYSTEM` disk. You will then use the copy for your normal work, keeping the original disk safely so that you can always make a new copy if the first is damaged in any way.

1.3.2 Initial startup procedure.

Having connected-up and switched on your PPC as described in section 1.3.1, you should now have the following message displayed on the screen:

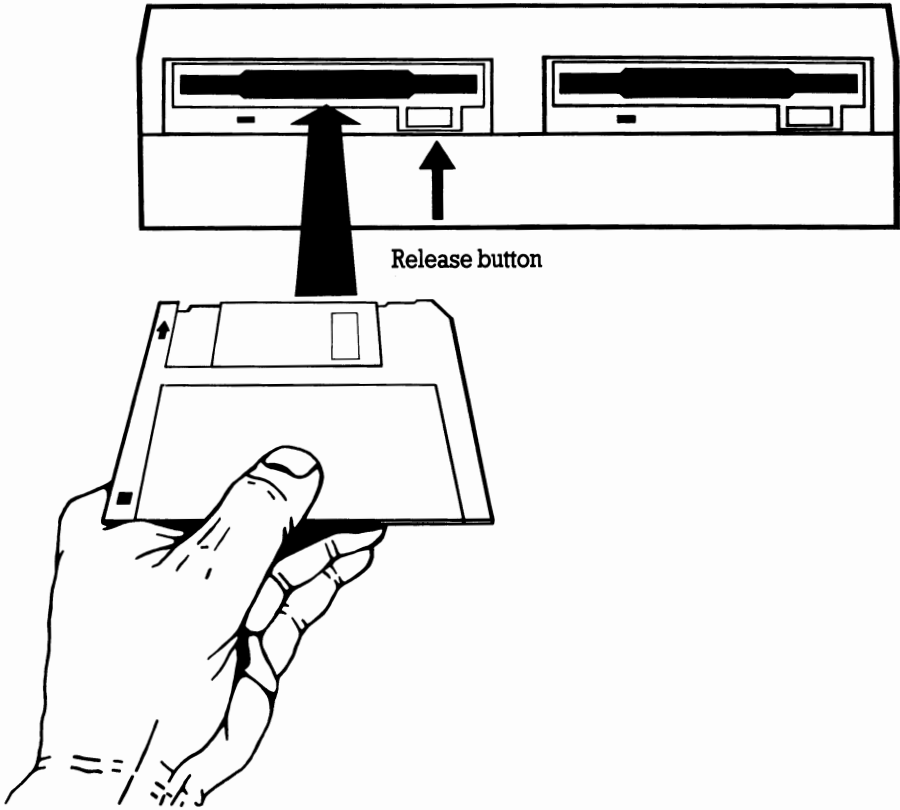
```
Insert a SYSTEM disk into drive A  
Then press any key
```

Take your `AMSTRAD PPC SYSTEM DISK` from its cover.



Hold the disk with the arrow uppermost and pointing toward the drive.

Gently insert the disk into the slot in your disk drive. If you have two floppy disk drives, insert it in the one nearest to the keyboard.



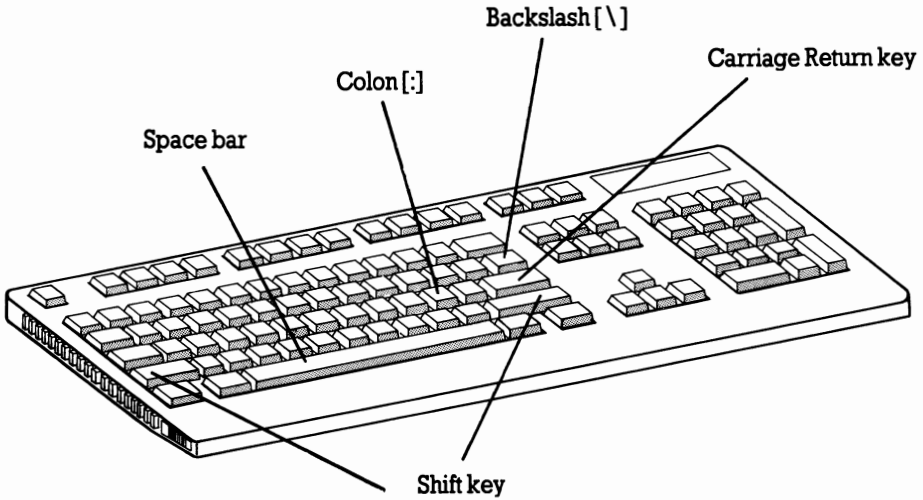
If the drive appears to be resisting the disk, stop. Check that there is no obstruction in the drive. Make sure that you have the disk the right way up, and try again. If you still have problems inserting the disk, consult your dealer. Whatever you do, don't force the disk into the drive.

Push the disk all the way into the slot until you feel it click into place.

To remove the disk from the drive, press the release button. The disk will slide partly out of the slot and can easily be withdrawn from the drive.

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

Press either the Space Bar or the Carriage Return key [↵] on the keyboard.



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

If you see a message similar to the following:

```
Wrong disk
Insert a system disk and press any key
```

press the release button and then withdraw the disk from the drive. Check that it is your AMSTRAD PPC SYSTEM DISK. If you had inserted the right disk, put it back in and press the Space bar. If it fails again, consult your dealer. If you had inserted the wrong disk, replace this disk in its cover and take the right disk out of its cover. Put this disk in the drive and press the Space bar.

After a short while a further message similar to the following will be displayed:

```
A>ECHO OFF
MS-DOS Version 3.30
With Scroll Lock on:
Right [Alt (Gr) ] - LCD text contrast toggle.
Right [Ctrl]     - LCD/CRT toggle.
```

```
A>
```

The LCD text contrast toggle is provided to give better monochrome resolution of colour displays. No single method of interpreting the colour information will give a satisfactory monochrome picture in all cases, therefore AMSTRAD have provided two alternative methods, which you can switch between to see which gives best results for any particular screen.

The LCD/CRT toggle allows you to switch the display instantly to and from an external monitor, which you can connect to the video socket on the rear of the PPC, provided that the dip switches are correctly set to select colour or monochrome, to suit the monitor to be used.

The line beginning with A > is the MS-DOS Command line. You could begin to use your system now, but if anything should happen to corrupt or damage your SYSTEM disk you would have to buy a new one in order to use the system at all. So your next step is to copy your SYSTEM disk; then you can use the copy for your normal use, keeping the original in a safe place and making further copies if the first one becomes unusable for any reason. It is permissible to make a copy of your AMSTRAD PPC SYSTEM DISK for use on your own PPC, but you must not transfer it to another person or machine except in accordance with the terms of the USER PROGRAM LICENCE AGREEMENT, as shown at the end of this manual.

NOW WORK THROUGH ONE OF THE FOLLOWING PROCEDURES TO MAKE COPIES OF YOUR SYSTEM DISK.

FOR A SINGLE DISK SYSTEM READ SECTION 1.3.2.1

FOR A DUAL DISK SYSTEM READ SECTION 1.3.2.2

1.3.2.1 Setting up a single disk system.

For a single disk system, leaving the SYSTEM disk in the drive:

At the A > prompt, type in `D I S K C O P Y` and press Carriage Return.

You will see the message:

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

Press the space bar and you will see the message:

```
Copying 80 Tracks.  
9 Sectors/Track, 2 side(s)
```

followed after a little while by the message:

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

Remove the SYSTEM diskette and insert a new diskette.

Press the space bar and the files from the SYSTEM disk will be copied onto the new disk. You may be prompted to insert the SOURCE (SYSTEM) and TARGET (NEW) disks again. Insert the disks as requested, until the message:

```
Copy another diskette (Y/N) is displayed.
```

Type [N] to end the command, remove the copy, label it and OPEN the write protect window, to protect it from accidental overwriting. Store the original PPC SYSTEM DISK in a safe place, away from excessive heat or magnetic fields.

NOW GO TO SECTION 1.3.2.3

1.3.2.2 Setting up a dual disk system

For a dual disk system, leaving the SYSTEM disk in the drive:

At the A> prompt, type in DISKCOPY A: and press [←].

You will see the message:

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

Insert a new disk in drive B. Press the space bar and the files will be copied from the SYSTEM disk in drive A to the new disk in drive B. When copying is complete, the message:

```
Copy another diskette (Y/N) is displayed
```

type [N] to end the command, remove the copy and label it. Store the original PPC SYSTEM DISK in a safe place, away from excessive heat or magnetic fields.

You can now follow the same instructions as for a single disk system, in section 1.3.2.3

1.3.2.3 Reset procedure and normal startup for floppy disk systems

With no disk in the machine, press the [Ctrl], [Alt], and [Del] keys at the same time. The [Ctrl] and [Alt] keys are next to each other towards the left hand side of the keyboard, while the [Del] key is next to the [Enter] key at the bottom right hand corner of the keyboard. Pressing all three together resets your computer. The computer will behave as if it had just been switched on and eventually the message:

```
Amstrad Portable PC 640K (V1.8) 09:00 on 10 January 1988
(c)1987 AMSTRAD plc

Last used at 08:59 on 10 January 1988

Insert a SYSTEM disk into drive A
Then press any key
```

Will be displayed. Insert your copy of the SYSTEM disk, press the space bar and the system will be loaded, displaying the message:

```
A>ECHO OFF
```

```
MS-DOS Version 3.30
```

```
With Scroll Lock on:
```

```
Right [Alt (Gr) ] - LCD text contrast toggle
```

```
Right [Ctrl      ] - LCD/CRT toggle
```

```
A>
```

If you have problems, repeat the procedure using the original disk and, if all goes well, make a further copy and try again with that. If you still experience difficulties, consult your dealer.

If the batteries are dead, the time and date messages will not appear. Instead, you will be reminded to set the time and date and fit new batteries. You are not obliged to do so; MS-DOS can be loaded and the PPC used, but the time and date used in any labels or messages will be incorrect. If the batteries are not replaced, the time and date will be correctly maintained, once set, until the PPC is switched off.

If you want to you can start using the MS-DOS operating system right away now. To do this turn to Part 2 of this guide: 'Using MS-DOS commands'. Alternatively, you may wish to begin with the MIRROR program, particularly if you are already familiar with MS-DOS. But before doing so it is a good idea to set the time of your PPC'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 shown in section 1.3.3.

From now on, if you run into difficulties you can reset the system by pressing the **[Alt]** **[Ctrl]** and **[Del]** keys together; inserting your copy of the SYSTEM disk when asked to do so. The system will reload and the A > prompt will be displayed. Resetting will not harm the system in any way, but anything which you have typed in will be lost, unless it has been saved to a file, and you will have to reload your program.

You can RESET without removing your floppy disks from the machine, but you should unload them before switching off and reload them after switching on, to avoid any risk of accidental corruption.

In future, whenever you switch on, insert your copy of the SYSTEM disk when asked; the system will be loaded from it and the A > prompt will be displayed. If you are a beginner you may find it helpful to study the notes on using the keyboard, in section 1.3.4, before going further. Remember, if you put a comma where a dot should be, or vice-versa, in a command, the command will not be accepted, or may be executed differently from the way you were expecting. Also \ and / have different meanings as do ; and : so take care over your typing and check that you have entered commands correctly.

1.3.3 Setting the internal clock and calendar.

Your PPC will automatically ask that you set the time and date whenever the SYSTEM disk is loaded and the clock requires setting. you can only set them after MS-DOS is loaded, because the commands which you use are MS-DOS commands.

At the DOS prompt, type TIME and press the Carriage 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 Carriage Return. If not, enter the new time as hh:mm:ss.dd

where hh represents the hour(00...23) mm the minute (00...59) ss the second (00..59) and dd the decimal parts of a second (00..99).

You need not enter fractions of a second, seconds, or even minutes, if you don't want to. The options you ignore will be set to zero.

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

TIME [↵]

16:00 [↵]

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

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

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

For example to set the date to December 20, 1988, type:

DATE and press [↵]

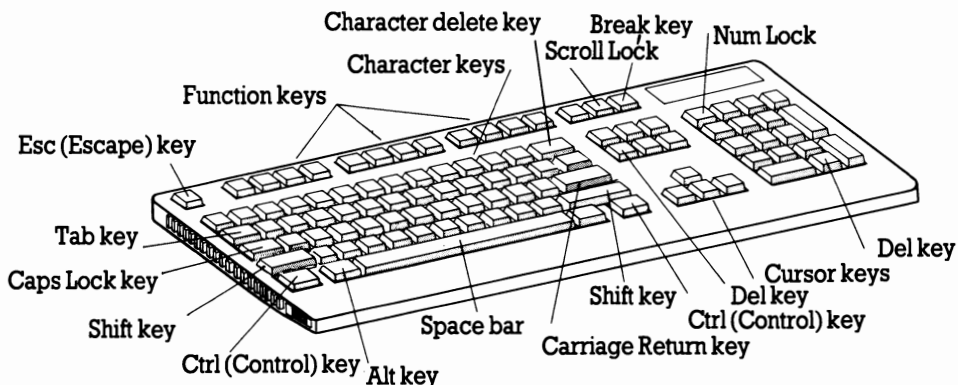
12-20-88 and press [↵]


The date which you have just entered is now set in the system. Once set, date and time will be kept correct so long as the internal batteries are serviceable, even when the system is switched off. If you keep the system connected to an external power source whilst replacing the batteries, the time and date will remain correct.

Note that the TIME and DATE commands are explained in much greater detail in Part 2.

The A> prompt is now shown on the display, indicating that you may enter any DOS commands.

1.3.4 Using the keyboard



Key	Use
Character keys	To enter lower case letters and the lower character written on a key where there are two. Used with [Shift] , to enter upper case letters and the upper character written on the key where there are two.
Function keys	Set up by the programs you run to have actions specific to the program you are running
	(known as the Space Bar) Enter a space character
[Shift] [Ctrl][Alt]	Used with other keys to give special characters or special computer codes (described elsewhere). Ctrl stands for Control; Alt stands for Alternative.
Cursor keys [↑] [↓] [→] [←]	Move the cursor on the screen in the direction shown by the arrow.
[Home] [End] [PgUp] [PgDn]	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.
[←]	Move the cursor to the beginning of the next line; send instructions to your PPC; reply to requests for information
[Tab]	Move the cursor to the next Tab Stop, inserting spaces as necessary
[←]	Delete the character to the left of the cursor
[Esc]	Used by programs, for example as the way you indicate that you want to stop using the program. Esc stands for Escape.
[Ins]	Used to swap between inserting characters into text already on the screen and overwriting existing text, and back again
[Num Lock]	Used to swap between using the keypad keys to control the cursor and using these keys as a numeric keypad, and back again
[Caps Lock]	Used to swap between the letter keys producing lower case characters without the [Shift] key and upper case characters with this key, and the opposite pairing. Pressed again to swap back again.

- [Scroll Lock]** Used by some programs to make the cursor keys move the text on the screen behind the cursor, rather than move the cursor over the text. Pressed again to return to normal cursor movement.
- [Break]** Used with **[Ctrl]** to abandon the program that is currently being run
- [PrtSc]** Used with **[Shift]** to print out an image of the screen on your printer. It must not be pressed if you don't have a printer connected to your PPC.
- [Del]** Used like del-backwards to delete the character immediately before the cursor. Also used with **[Ctrl]** and **[Alt]** to reset your PPC.

Note:

Not all programs will use the keys in the same way. For example, [←] 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]**, **[PgDn]**, **[Tab]**, **[Ins]** and **[Break]** keys may have no effect.

1.4. ABOUT DISKS

1.4.1 FLOPPY DISKS

Included with your Amstrad PPC is a 3½ inch floppy disk containing the MS-DOS operating system and utilities. The PPC 640 is supplied with a further disk, holding the MIRROR communications program. As you use your PPC more, you will want to buy further programs on floppy disks and you will need to buy a number of new blank floppy disks - for storing data, for storing duplicate copies of the program you buy, and even your own programs.

This section describes what type of floppy disks you should use in your AMSTRAD PPC and how to handle them, also how to make backup copies of your disks.

Because disks can become damaged, it is good practice to keep 'security' copies of as many of your disks as you can, starting with the disk supplied with your PPC.

Note:

Some of the programs you buy may be protected against copying and you won't be able to make direct copies of these disks. Refer to the program's own user guide for details of how to make security copies of these programs.

The instructions for copying programs from one disk onto a fresh disk are given in section 1.7.1.

1.4.1.1 What type of disks?

The AMSTRAD PPC's built-in floppy disk drives can only accept 3½ inch floppy disks.

Floppy disks are themselves thin and circular (as their name suggests). They are perhaps the most sensitive parts of your computer system and so they are enclosed in a square plastic outer casing to protect them.

There are many different kinds of computer disk, so when buying blank disks or programs on disk you should check that you buy the right sort.

The disks you need are:

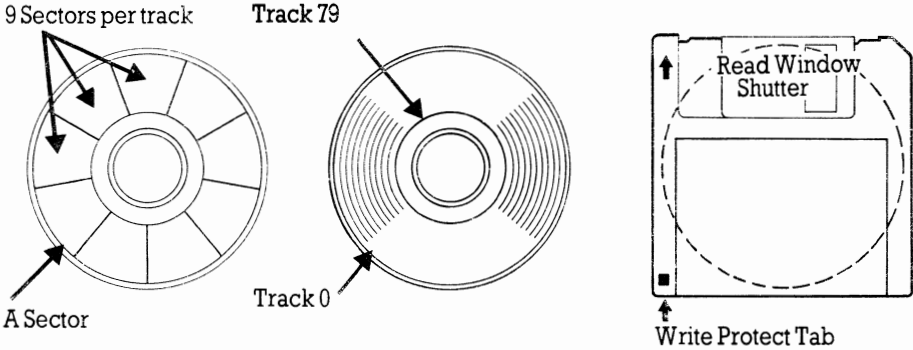
- Double-sided - 720 KB
- 135 tpi - Soft-Sector

We recommend you use labelled disks from any leading manufacturer. Using cheap, unlabelled disks can be a false economy and should be avoided. Data transfer between such disks can be unreliable and, in the worst case, their use can damage your disk drives and possibly all the other disks you try using in the drives.

Note:

Any blank disks you buy need to be prepared before you can use them to store any programs or data. This process is called formatting. It is done automatically when you copy a disk onto a new blank disk, but you have to format specially any new disk you want to have ready for storing the data files your programs produce. How to format a disk is described in section 1.6.2 (using an MS-DOS command).

The important features of a disk

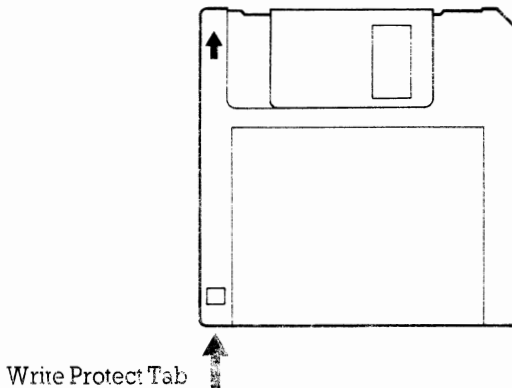


The first thing to notice about a $3\frac{1}{2}$ inch floppy disk is that it has an upper and a lower side. The upper side has the manufacturer's label on it and a small arrow imprinted in the plastic casing. On the lower side can be seen the metal driving hub, in the centre, and a plastic 'write protect' shutter in one corner. Disks are always put into the AMSTRAD PPC's drives with the arrow uppermost and pointing away from you.

The other features are:

The Reading Window: This window is the gap in the packaging through which the computer reads and writes data on the disk. It is protected by a metal shutter which you should never open or interfere with. Even the grease on your hands can spoil the data stored on the disk.

The Write-Protect window: If the shutter is positioned to leave the window CLOSED, your AMSTRAD PPC is ABLE to both read data from the disk and WRITE data to it. If the Write-Protect window is OPEN, the PPC will be able to read data from the disk but it WON'T be able to WRITE data to it. Such a disk is described as write protected.



Well written programs should warn you if you attempt to write to a protected disk; but some may simply fail, so be careful to ensure that disks which you want the program to write to have the window closed.

Tracks and Sectors: These are the divisions into which the data storage area of the disk is divided so that information can be stored on the disk. The tracks are numbered from 0 to 79 and the sectors from 1 to 9.

Most of the time you use your PPC, you have no need to know how the data storage area is divided up. We have included this information because if your PPC encounters a problem in reading or writing data in a particular part of the disk, it may tell you where the problem is by giving its track and sector numbers. Alternatively, you can find out where the problem is by 'Verifying' the disk. (How to verify a disk is described in section 2.4 (using the MS-DOS operating system).

If you experience such problems repeatedly in the same track and sector of a disk, this is a sure sign that the disk has been damaged and we would advise you to copy what information you can onto another disk and then stop using the damaged disk.

If you find that a number of disks are becoming damaged in this way, check with your dealer that your disk drive isn't damaged.

1.4.1.2 How to handle disks

The thing to remember about your disks is that they are highly sensitive - to scratches, to dust, to temperature, to moisture and to magnets. So:

- * Don't try to force open a disk's plastic casing or poke anything into it
- * Don't store disks anywhere they could get damp or hot or cold
- * Don't store disks anywhere near a magnet: that includes the magnets within such pieces of equipment as the loudspeakers of your audio equipment, your telephone or your TV
- * Never put disks into your PPC's floppy disk drives before you switch on and never turn your PPC off with any disk still in a floppy disk drive
- * Never take a disk out of a drive while your PPC is reading from it or writing to it (You can tell if your PPC is writing or reading a disk because the green, drive indicator light on top of the PPC will either be fully on or flashing on and off.)

Treat floppy disks with due respect and you will get good service from them.

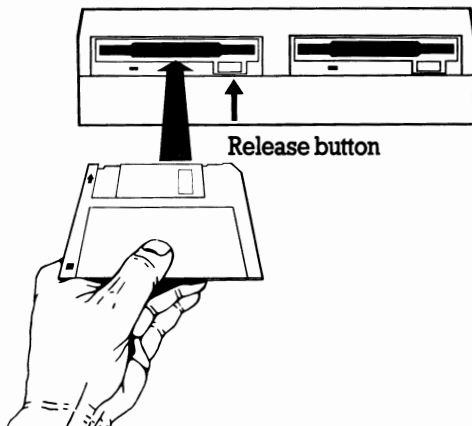
1.4.1.3 Inserting a disk into a floppy disk drive

Note:

Never insert a floppy disk into a drive before switching your PPC on.

- a. Check that there's nothing in the drive.

- b. Withdraw the disk you want to use from its cover.
- c. Hold the disk with the arrow uppermost and pointing away from you.
- d. Insert the disk fully into the drive slot. The disk should simply slide in and latch into place. If the drive appears to be resisting the disk, stop and try again. If you continue to have problems inserting the disk, consult your dealer. Whatever you do, don't force your disk in.



Note: If the instructions tell you to put a disk in Drive A, this means your floppy disk drive -or the drive nearest the keyboard if you have two floppy disk drives. The floppy disk drive furthest from the keyboard on a two-drive PPC is referred to as Drive B.

1.4.1.4 Releasing a disk from a floppy disk drive

Note:

Never release a disk from a drive while your PPC is either reading data from the disk or writing data on it. Not only could the data on the disk be damaged but the disk itself could suffer. You can tell if your PPC is using the disk because the green drive indicator light, on the top of the PPC, will be either fully on or flashing on and off.

Always release all disks from your floppy disk drive(s) before switching off.

- a. Press the release button, which will cause the disk to slide a little way out of the slot.
- b. Grip the edge of the disk with your fingertips and withdraw it from the drive slot.
- c. Put the disk back in its cover and store it safely, away from heat or strong magnetic fields.

Note:

If, as sometimes happens, you accidentally knock the disk back in a little as you go to withdraw it, making it difficult to grip, simply push it fully in and then release it once more.

1.5 RUNNING PROGRAMS

1.5.1 WHAT YOU CAN USE YOUR AMSTRAD PPC FOR

In buying an AMSTRAD PPC, you have acquired a powerful 16-bit microcomputer. You probably already have a rough idea what you are going to use it for, but now is the time to look at what your PPC can do in a little more detail. It might be that it can do more than you originally thought.

If you plan to use your AMSTRAD PPC at the office, you might use it to:

- prepare letters and reports
- keep accounts and prepare invoices
- keep all the information you need to refer to available at your fingertips
- generally help you to carry out routine tasks

If you plan to use your AMSTRAD PPC at home, you might use it to:

- prepare letters
- keep your address book and phone list up to date
- computerise your home accounts; help you budget ahead
- write and develop your own programs
- play computer games

The AMSTRAD PPC is supplied to you with 'system' programs, that is the 'tools' needed to use your computer.

Also provided is PPC ORGANIZER, an integrated desktop utilities program, which provides the following facilities:

- Database of address cards with phone list
- Database of memo cards
- Letter writing and Wordprocessing
- Calculator
- Autodialling (PPC 640 only)
- Appointments Diary and Alarm

Full details of this program are contained in the separate booklet 'PPC Organizer User Instructions', also supplied with your PPC.

The PPC 640 is supplied with an integral modem and the MIRROR II communications software. At the beginning of section 3 you will find a quick introduction to the features of the modem and an illustration of its use.

An extremely wide range of programs can be run on the AMSTRAD PPC, giving you a good choice not only of types of program to run but also of which program of its kind to use - which word processing package, which spreadsheet, which database, which programming language etc. etc.

You don't even have to buy programs specifically described as being for use on the AMSTRAD PPC. The AMSTRAD PPC's compatibility with the IBM PC and PC compatibles means you can use programs written for these machines and running under either MS-DOS or PC-DOS.

Nor do you need to buy programs especially to use them on your AMSTRAD PPC. If you have an IBM PC or a PC compatible at the office, you might well be able to just put one of your IBM PC disks straight into your AMSTRAD PPC and use it exactly as if it were the office PC (provided you are not breaking any copyright or licence agreements in doing so).

What you need to check is that the programs you use are:

- * written to run under PC-DOS or MS-DOS
- * written for an IBM PC or a PC compatible

and

- * supplied on 3¹/₂ inch, 720 KB floppy disks.

The vast majority of these programs will run perfectly first time on the AMSTRAD PPC. Other programs may run but won't always produce the results you expect. They may not control the screen properly or they may not respond correctly to the keyboard.

Such programs will need to be 'installed' on your AMSTRAD PPC. Installing a program is not a difficult job in most cases but newcomers to computing are advised to choose programs that don't need installing if possible. Your computer dealer will be able to advise you on this. If you do need to install a program, there is advice on how to go about this in appendix A. This appendix also describes how to prepare disks holding only the programs you need while you are using a particular application program, for example your word processor or your spreadsheet.

The rest of this section is given over to describing how to get started using your AMSTRAD PPC in two ways:

- * running a popular DOS program, by using the DOS command line (as described in section 2 of this manual)
- * using a program on a disk set up to just load and run a 'Turnkey' program

Note:

- i) For all programs other than Turnkey programs, you need to have loaded a DOS operating system before you can start using the program. Turnkey programs are set up so that this software is loaded automatically.
- ii) For PPC 640 owners, running the MIRROR II communications program is fully described in section 3 of this manual.

IMPORTANT: WORK IF POSSIBLE WITH COPIES OF DISKS RATHER THAN WITH THE ORIGINALS. KEEP THE ORIGINALS SAFELY STORED AWAY AS YOUR MASTER COPIES FOR USE IN CASE OF ACCIDENT TO MAKE FURTHER COPIES FOR YOU TO USE.

1.5.2 Running a popular DOS program

The instructions given in this section will help you run any of the major programs used on the IBM PC - programs whose names often appear in magazines for the business computer user. As you follow these instructions, refer to the program's own user guide. This will give you more detailed instructions on what to do.

The disks you have just bought are both valuable and vulnerable to accidents. So the first thing to do is to make copies of these disks. Then store away the original disks for use only to make new copies from in case of accidents.

Note:

Some programs are copy protected - you can't copy these entirely. However, you may be able to copy all but a part of this software - if so, how to do this will be described in the program's user guide.

Find out from the program's user guide whether the program generates any data files. Spreadsheet programs, for example, certainly do. If it does, you may need to prepare (that is, format) a new blank disk specially to hold these data files. Alternatively, you may have enough free space on one of your other disks to store the new program's data files there.

Next, discover from the program's user guide:

- the full name of the program you want to run
- which disk it is stored on (if the software is stored on more than one disk)
- which directory or folder it is stored in.

The program will have a two-part name, the parts being separated by a full stop. The second section of the name will be either BAT, COM or EXE. If the user guide doesn't say which directory or folder the program file is in, assume it is in the main or 'root' directory of the disk.

Before you go any further, check whether the program makes use of any of the DOS external commands or 'utilities'. If it does, discover from the user guide which drive it expects to find these on because you will need to put the disk holding the commands in this drive.

Notes:

- i) MS-DOS commands are on the PPC SYSTEM disk. Unless the user guide has been written specifically for the AMSTRAD PPC, it will probably tell you to look for these commands on the wrong disk.
- ii) The program's user guide may refer to commands or utilities that have the same filename as a command on your SYSTEM disk but have a different filetype. In particular, it may give the filetype COM when the command on your SYSTEM disk has the filetype EXE. The files on your SYSTEM disk are the ones you need. If you copy any of these files, make sure you give the copy the filename and filetype used on your SYSTEM disk.
- iii) Make sure you know what each of the disks supplied with your program is called, and be careful to check whether you should have a program disk or your SYSTEM disk loaded at any one time. You won't harm the disks, provided you are careful not to close the write protect window on any disk which shouldn't be overwritten, but you can get very muddled if you don't work carefully and systematically.

1.5.3 Using a 'Turnkey' program

A 'Turnkey' program is one stored on a disk together with the system programs in such a way that your PPC will automatically run the program immediately after it has read the operating system software into its memory.

Two ways of running such a program are described here. The first of these describes how to run the program as the first thing you do after switching on; the second describes how to run the program when you have already been working on your machine.

Notes:

i) Some program user guides assume that your PPC has two floppy disk drives. Don't worry if you have a single drive system: just type in exactly what it says for two floppy disk drives called Drive A. Your PPC will ask you to insert the disk for Drive A when it wants programs or data you have told it are on Drive A, and it will ask for the disk for Drive B when it wants programs or data that you have told it are on Drive B.

ii) Make sure you know what each of the disks supplied with your program is called, and be careful to check whether you should have a program disk or your SYSTEM disk loaded at any one time. You won't harm the disks, provided you are careful not to close the write protect window on any disk which shouldn't be overwritten, but you can get very muddled if you don't work carefully and systematically.

iii) If anything happens as you work through these instructions that is not explained here, turn to the 'troubleshooting' section, appendix G.

Switch on and run

Start with

- the plug out of the wall outlet
- the power switch on the top of the PPC in its EXT.ON position
- no disk in any floppy disk drive.

Work through the Startup procedure given in section 1.3, using the program disk as the Startup disk. Follow any additional instructions either in the program's user guide or that appear through messages on the screen.

While the PPC is reading software from your disk, you will see the green, drive indicator, light, on top of the PPC, go on and off a few times. When it has finished, the Display will be showing the first screen display of your program.

Turn to the program's own user guide for details of how to carry on.

Running the program when the PPC is already on

Put the disk holding your program into Drive A (the drive nearest the keyboard if you have two) and push it fully home.

Hold down the **[Ctrl]** and **[Alt]** keys and press the **[Del]** key. Your PPC then reads the software from this disk. You will see the green, drive indicator, light, on top of the PPC, go on and off a few times while this is happening.

When it has finished, the Display will be showing the first screen display of your program. Turn to the program's own user guide for details of how to carry on.

Note:

Some user guides describe, for example, typing in the date before you really start working on the program. Beware: the descriptions they give may not be based on MS-DOS 3.3 but on, for example, PC-DOS 2.0. Be sure to check in this manual for details of how you should enter the information.

1.6. USING MS-DOS TO ORGANISE YOUR DISKS

This section gives step by step instructions on each of the tasks involved in keeping your disks and the files stored on them ready for use. The contents are as follows:

- 1.6.1 Copying a disk
- 1.6.2 Preparing a new blank disk (Formatting)
- 1.6.3 Copying a file
- 1.6.4 Deleting a file
- 1.6.5 Renaming a file
- 1.6.6 Finding out the size of a file
- 1.6.7 Finding out how much unused storage space there is on a disk
- 1.6.8 Using the Editor to create or change a text file

IMPORTANT:

WORK IF POSSIBLE WITH COPIES OF DISKS RATHER THAN WITH THE ORIGINALS. KEEP THE ORIGINALS SAFELY STORED AWAY AS YOUR MASTER COPIES FOR USE IN CASE OF ACCIDENT TO MAKE FURTHER COPIES FOR YOU TO USE.

1.6.1 Copying a disk

The process of copying a disk onto a fresh disk makes the second disk a 'photocopy' of the first. Immediately after the copy has been made, the two disks are identical in every way, but subsequent processing of either disk will stop them from being identical.

Note:

- i) If you want to put the copy on a new blank disk, you don't have to format the new disk (as described in section 1.6.2) before you start to make your copy.
- ii) If you copy a Startup disk (that is, a disk that you use immediately after switching your machine on) you create another disk that can be used as a Startup disk.

The steps to copying a disk are as follows:

- a. Display an A> system prompt.

If you are not sure how to do this, put your SYSTEM disk in Drive A (the drive nearest the keyboard, if you have two) hold down **[Ctrl]** and **[Alt]** and press **[Del]**

- b. Make the root directory the current directory.

If you have just reset your PPC, you need not do anything. Otherwise, you may need to type:

```
CD \      [←]
```

to return to working on the root directory.

- c. Type:

```
DISKCOPY A: B:    [←]
```

What happens next depends on whether your PPC has one or two floppy disk drives.

*** If you have a single-drive PPC**

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

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

d. Remove the MS-DOS Startup disk from the drive and insert the disk you want to copy.

Press the release button to release the disk, then withdraw it completely from the drive and replace the disk in its cover.

Insert the disk you want to copy in the drive and push it fully home.

e. 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 - n tracks
Sectors/track, 2 side(s)
```

followed by:

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

f. Remove the disk from the drive and insert the disk you want to store the copy on.

Press the release button, withdraw the disk completely from the drive and replace the disk in its cover. Insert the disk you want to store the copy on in the drive.

g. 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 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)?
```

h. Type [Y] if you want to copy another disk; type [N] to leave DISKCOPY.

If you have a two-drive PPC

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 . . .
```

d. Remove the disks currently in the drives and insert the disk you want to copy in drive A (the drive nearest the keyboard) and the disk you want to store the copy on in drive B (the drive at the rear of the PPC).

To remove a disk, press the release button and then withdraw the disk completely from the drive. Replace the disks in their covers.

Insert the disk you want to copy and the disk you want to store the copy on in their appropriate drives and push them fully home.

e. 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 - n tracks
m Sectors/track, 2 side(s)
```

If you are storing the copy on a new blank disk, or on a disk which has a different format to your Source disk, you will see the message:

```
Formatting while copying
```

and finally the message:

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

f. Type [Y] if you want to copy another disk; type [N] to leave DISKCOPY.

1.6.2 Preparing a new disk for use - Formatting

The storage area of a new disk needs to be marked out in sections electronically before you can use the disk to store any programs or data.

This process is called formatting.

When you copy a disk onto a new blank disk, the new disk is automatically formatted for you as part of the copying process. You only have to remember to format the new disk when you want to have it ready to store, for example, the data files which your spreadsheet program is about to produce.

The process can also be used with an old disk the data on which has become corrupted. If the disk hasn't been physically damaged, formatting the disk can reclaim all its storage space. However, before you do this, do 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.

Note:

The method described here prepares a disk for use simply to store programs and data. If you want to prepare a disk for use as a Startup disk (that is, to insert immediately after you switch your PPC on), you should either turn to section 2.6.2 or copy an existing Startup disk and delete all the files stored on it.

The steps you need to take are as follows:

a. Display an A > system prompt.

Return to Drive A as the default drive. If you are not certain how to do this, reset your machine - instructions for this are given in section 1.3.2.3. If you have yet to switch on your PPC, go through the Startup procedure described in section 1.3.1

b. Place your SYSTEM disk in Drive A (the drive nearest your keyboard if you have two).

This disk will already be in the drive if you have just completed the startup or reset procedure.

c. Make the root directory the current directory.

If you have just switched on or reset your PPC, 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.

d. Type:

```
FORMAT A:  [↵]
```

After a short while, you should see the following message:

```
Insert new diskette for drive A:  
and press RETURN when ready
```

e. Remove the Startup disk from Drive A and insert the disk you want to format.

To remove a disk, press the release button to release the disk and then withdraw it completely from the drive. Replace the disk in its cover. Put the disk you want to format in the drive and push it fully home.

f. Press the [↵] key.

The message: `Head:n Cylinder:m` 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)?

g. Type [Y] [↵] if you want to format another disk;

type [N] [↵] to leave `FORMAT`.

1.6.3 Copying a file

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 folder (or directory) as the original or in a different folder, often 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 folder (or directory), so if you want to copy files from separate folders, you must repeat the process for each folder 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:

a. 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 (that is, a file name covering a number of files by including the wildcard characters * and ? - see section 1.7.5) and you don't want to store the copies in the same directory as the original files. If you want to make duplicate copies of files in the same directory you have to copy the files one at a time.

b. 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 file-name`. 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

(Turn to section 1.7.7 if you need extra help in working out what to write down.)

c. 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 don't 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 1.6.7.

d. 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) which the source file(s) already have, or

```
drive:\directory\file-name
```

if you want to use different file name(s).

e. 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.

1.6.4 Deleting a file

Deleting files removes the information stored in them from your disk: this information cannot be recovered. You can delete files either one at a time or in a group (provided they are all in the same folder/directory).

The steps are as follows:

a. 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 (that is, a file name covering a number of files, by including the wildcard characters * and ? - see section 1.7.5).

b. 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 file-name. 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

(Turn to section 1.7.7 if you need extra help in working out what to write down.)

c. 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 do want all the files to be deleted;

Type [N] [↵] and then redo your command line if you hadn't intended to delete all the files.

1.6.5 Renaming a file

Files do not have to keep the name they were given when they were created. In most cases, you can give a file a new name any time you choose.

The new name you choose should:

- be different from any other file or folder name in this folder
- suit the needs of the programs you are going to use to process the file (refer to the programs' 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, but you can use some other characters as well (see section 1.7.4).

The steps are as follows:

a. 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 : \ D I R 1 \ S U B D I R \ M Y F I L E . T X T

if SUBDIR is listed in the DIR1 directory, which is itself listed
in the Root directory on Drive A

(Turn to section 1.7.7 if you need extra help in working out what to write down.)

b. Jot down just 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

c. Type in the appropriate REN command. The command you need is:

R E N 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 doesn't 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.

1.6.6 Finding out how large a file is

a. 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 : \ S U B D I R \ M Y F I L E . T X T

if SUBDIR is listed in the Root (principal) directory on Drive A

A : \ D I R 1 \ S U B D I R \ M Y F I L E . T X T

if SUBDIR is listed in the DIR1 directory, which is itself listed in the Root directory on Drive A

(Turn to section 1.7.7 if you need extra help in working out what to write down.)

b. Type:

D I R 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.)

1.6.7 Finding out how much unused storage space there is on a disk

a. Insert the disk you want to examine in Drive A (the drive nearest the keyboard if you have two) and push it fully home.

b. Type:

```
CHKDSK A: [↵]
```

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.)

1.6.8 Using the Editor to create or change a text file

The AMSTRAD PPC is supplied with a screen-based Text Editor called RPED, which is very useful for editing small text files. As 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 cursor 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.

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 (that is, 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, or create a new file, RPED needs details of where the existing file is stored, or the new file 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

a. Display an A> system prompt.

Switch on and load MS-DOS, by following the Startup procedure given in section 1.7.1.

If you are already using MS-DOS on your PPC, finish what you are doing then return to an A> system prompt and type:

```
CD \ [↵]
```

b. Type:

```
RPED [↵]
```

After a short while, the following should be seen on the screen. This is RPED's opening screen:

```
This screen editor is for small files (up to 750 lines) and uses
normal Cursor, Page, Home, End, Insert and Delete keys.
```

```
f1 = EditExisting File
```

```
f2 = Re-edit Previous File
```

```
f3 = Create New File
```

```
f4 = Quit
```

1.7. FUNDAMENTALS OF USING YOUR AMSTRAD PPC

This chapter is a reference chapter in which you can quickly look up standard procedures like the AMSTRAD PPC's Startup procedure, and other standard information such as how to construct a filename. The contents are as follows:

- 1.7.1 The Startup procedure
- 1.7.2 Switching off
- 1.7.3 Resetting your PPC
- 1.7.4 About filenames
- 1.7.5 Using wildcards to specify a number of files
- 1.7.6 About folder and directory names
- 1.7.7 Specifying a path
- 1.7.8 Cleaning your PPC

1.7.1 The AMSTRAD PPC Startup procedure

If anything happens as you work through this Startup procedure that is not explained here, turn to the 'troubleshooting' section, appendix G

Note:

In this Startup procedure, you can use

- the SYSTEM disk supplied with your AMSTRAD PPC (preferably a copy of this)
- any other disk with Operating System software on it. This includes commercial program disks that have been set up so that you can just turn on your computer and run the programs - so-called 'Turnkey' programs.

a. Start with

- no external power supply connected to the PPC
- the power switch on the top of the PPC in its EXT.ON position
- no disk in any floppy disk drive

b. Either connect and plug in one of the three permitted external supplies (see Introduction) or switch to BATT.ON to run on the internal batteries.

Your PPC then goes through a built-in system check. If all is well, a message similar to the following will shortly appear on the screen:

```
Amstrad Portable PC 640K (V1.8)  09:00 on 10 January 1988
(c)1987 AMSTRAD plc
```

```
Last used at 18:00 on 09 January 1988
```

```
Insert a SYSTEM disk into drive A
Then press any key
```

Note:

The first time and date displayed is the current setting of the on-board clock. The second time and date records when you last finished using the operating system supplied with your machine. If you use a different operating system on your AMSTRAD PPC, the time and date shown will record when you last switched your PPC on.

If the batteries are dead, the time and date messages will not appear. Instead, you will be reminded to set the time and date and fit new batteries. You are not obliged to do so; MS-DOS can be loaded and the PPC used, but the time and date used in any labels or messages will be incorrect. If the batteries are not replaced, the time and date will be correctly maintained, once set, until the PPC is switched off.

Use of the TIME and DATE commands is described in section 1.3.3

Note:

These commands cannot be used until MS-DOS is loaded.

If something is holding down keys on the keyboard, the following message will appear:

Check Keyboard

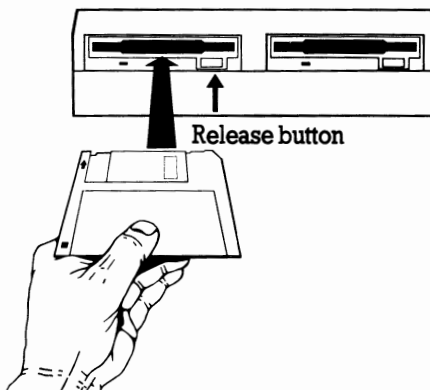
Clear everything off the keyboard and check that none of the keys are stuck down.

c. Insert a STARTUP disk into Drive A.

- To load MS-DOS, insert your AMSTRAD PPC SYSTEM disk - To load a 'Turnkey' program, insert the program disk

IMPORTANT: WORK IF POSSIBLE WITH COPIES OF DISKS RATHER THAN WITH THE ORIGINALS. KEEP THE ORIGINALS SAFELY STORED AWAY AS YOUR MASTER COPIES FOR USE IN CASE OF ACCIDENT TO MAKE FURTHER COPIES FOR YOU TO USE.

Insert the disk into the slot in your disk drive. If you have two disk drives, insert it into the slot of the drive nearest the keyboard, and push it fully home until it clicks into place.



d. Press any key on the keyboard. The space bar or the [←] key is usually the most convenient choice.

The AMSTRAD PPC then reads the system software from this disk. You will see the green drive indicator light on top of the PPC go on and off a few times while this is happening.

The screen will clear after a short while and the following additional messages will be displayed,

```
A>PATH \;  
A>KEYB US 437  
A>ECHO OFF  
Current date is Tue 1-01-1980  
Enter new date (mm-dd-yy):
```

and, after you have entered the date and time;

```
MS-DOS Version 3.30  
With Scroll Lock on:  
Right [Alt (Gr)] - LCD text contrast toggle.  
Right [Ctrl] - LCD/CRT Toggle.  
A>
```

if you have loaded MS-DOS.

If you have loaded a 'turnkey' program, its documentation should tell you what message to expect and how to continue.

The LCD text contrast toggle is provided to give better monochrome resolution of colour displays. No single method of interpreting the colour information will give a satisfactory monochrome picture in all cases, therefore AMSTRAD have provided two alternative methods, which you can switch between to see which gives best results for any particular screen.

The LCD/CRT toggle allows you to switch the display instantly to and from an external monitor, which you can connect to the video socket on the rear of the PPC, provided that the dip switches are correctly set for a colour or monochrome monitor.

1.7.2 Switching off

a. Check that your PPC isn't reading any data from your disks or writing data to them. If either of the green, drive indicator lights on top of the PPC is either fully on or flashing on and off, the AMSTRAD PPC is reading or writing data to a disk drive.

b. Press the release button on each floppy disk drive. This automatically releases the disk(s) from the drive(s).

c. Withdraw the disk(s) from the drive(s) and replace them in their covers.

d. Only then, unplug the external power supply or, if you have been using the internal batteries, switch the power selector switch from BATT.ON to BATT.OFF.

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

1.7.3 Resetting your PPC

[Ctrl] + [Alt] + [Del]

Resetting your PPC 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.

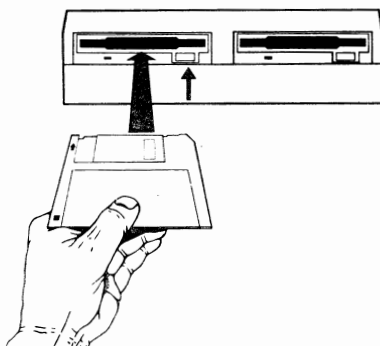
* You should not reset your PPC when there is important information currently held in its memory without considering whether this information can be saved to disk first. Resetting your PPC clears its memory. Any information that hasn't been saved onto disk will be lost.

The steps to take are as follows:

- a. Remove the disk(s) currently in the drive(s). To remove a disk, press the release button to release the disk and then withdraw the disk completely.
- b. Hold down both the **[Ctrl]** and **[Alt]** keys and press the **[Del]** key.
- c. Insert your Startup disk into Drive A (the drive nearest the keyboard, if you have two).
 - To load MS-DOS, insert your AMSTRAD PPC SYSTEM disk
 - To load a 'Turnkey' program, insert the program disk

IMPORTANT: WORK IF POSSIBLE WITH COPIES OF DISKS RATHER THAN WITH THE ORIGINALS. KEEP THE ORIGINALS SAFELY STORED AWAY AS YOUR MASTER COPIES FOR USE IN CASE OF ACCIDENT TO MAKE FURTHER COPIES FOR YOU TO USE.

Insert the disk into the slot in your disk drive. If you have two disk drives, insert it into the slot of the drive nearest the keyboard, and push it fully home until it clicks into place..



d. Press any key on the keyboard. The space bar or the [←] key is usually the most convenient choice.

The AMSTRAD PPC then reads the system software from this disk. You will see the green, drive indicator light on top of the PPC go on and off a few times while this is happening.

The screen will clear after a short while and the following message will be displayed,

```
MS-DOS Version 3.30
With Scroll Lock on:
Right [Alt (Gr)] - LCD text contrast toggle
Right [Ctrl]      - LCD/CRT toggle

A>
```

if you have loaded MS-DOS.

If you have loaded a 'turnkey' program, its documentation should tell you what message to expect and how to continue.

The LCD text contrast toggle is provided to give better monochrome resolution of colour displays. No single method of interpreting the colour information will give a satisfactory monochrome picture in all cases, therefore AMSTRAD have provided two alternative methods, which you can switch between to see which gives best results for any particular screen.

The LCD/CRT toggle allows you to switch the display to and from an external monitor, which you can connect to the video socket on the rear of the PPC, provided that the dip switches are correctly set for a colour or monochrome monitor.

1.7.4 About filenames

The rules about what you can call a file are laid down by a computer's operating system, which on the AMSTRAD PPC is MS-DOS. Under both these operating systems, 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 this manual, though 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 characters you can use are given in columns 3 to 8 on the lefthand page of the table in appendix C. However, the following special characters can't be used:

```
< > = | / [ ] , . : ; \ + { } ! * ? " ' m
```

and you may not put spaces into a filename.

For example, the following filenames are valid:

```
A AAAAAAAA 123
```

The following filenames are invalid:

A 2 3 [4] Includes one or more special characters

AAAAAAAAAA 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 (that is, a...z) are exactly the same to MS-DOS as upper case letters (that is, A...Z). For example, the filename C a t 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 can't 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 BASIC programs have the filetype BAS

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 be used for another file in the same folder or 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.)

1.7.5 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.

1.7.6 About folder and directory names

Directories and folders are the same thing - the groups of files that you can use to make your files more manageable. Thus the discussion below about what names folders can have applies directly to choosing directory names. The names that can be given to folders on the AMSTRAD PPC are like the names that can be given to files. Again the name is split into two parts - a Name and a Type. In practice, the Type part of the name is rarely used.

The Name is any combination of 1 to 8 characters, starting with one of the letters of the alphabet or a number. The characters you can use are given in columns 3 to 8 on the lefthand page of the table in appendix C. However, the following special characters can't be used:

< > = | / [] , . : ; \ + { } ! * ? " m

and you may not include any spaces in the name.

For example, the following names are valid:

A AAAAAAAAA 123

The following names are invalid:

A 2 3 [4]	Includes one or more special characters
AAAAAAAAAA	Too many characters

Often just the letters of the alphabet A...Z and the digits 0...9 are used.

Notes:

- i) Lower case letters (that is, a...z) are exactly the same to MS-DOS as upper case (that is A...Z). Cat, for example, is exactly the same as cat or CAT.
- ii) You must not use the names AUX, CON, COM1, COM2, LPT1, LPT2, LPT3, LST, NUL and PRN.

The Name will usually be chosen to remind the user about the files the folder holds but this doesn't have to be so.

The Type is up to three characters long and again the special characters and spaces can't be used. There are no conventions governing the combination of characters used.

A folder can be given any valid name provided it is not already the name of a folder in the same directory.

Folder names are used to describe paths to the folder holding the files you want to use. How this is done is described below.

1.7.7 Specifying a path

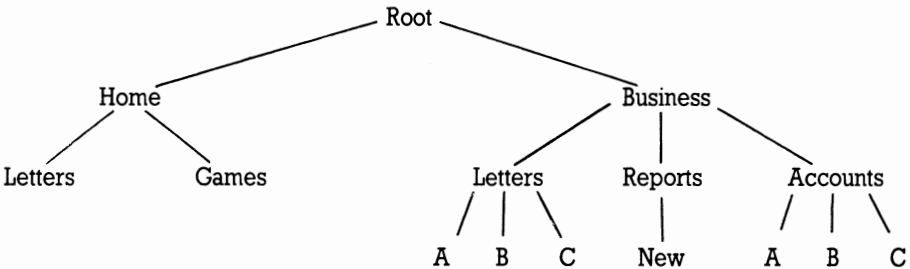
A 'Path' is used to direct MS-DOS to a particular folder on a DOS disk. It is a list of the folders between a point the operating system recognises and the new directory.

The starting point for the path is:

- either the root directory on the drive
- or the directory (folder) you are currently working with on the drive

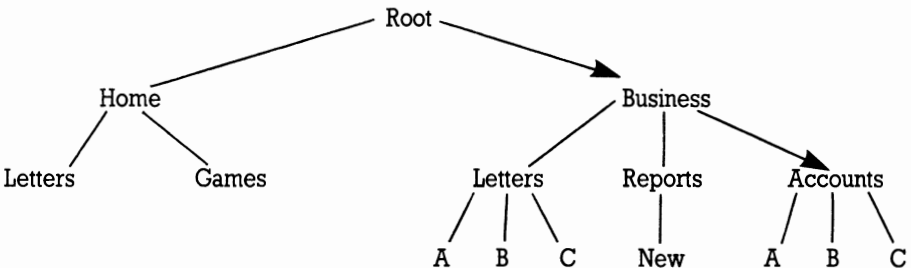
You can use the path from the Root directory regardless of which directory you are currently working with.

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`

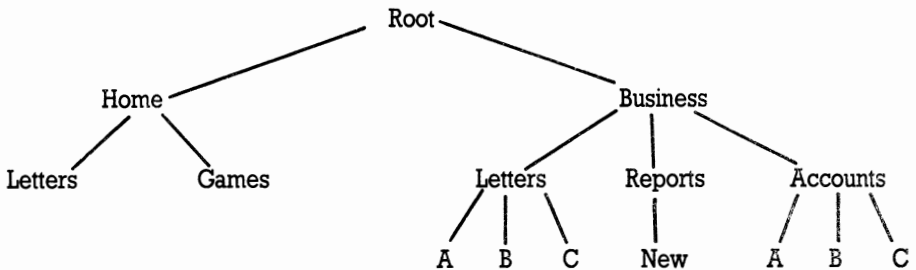
When you come to use this path, for example to specify a file in the directory, you precede it by a backslash (to tell the operating system that the path starts at the Root directory). If you are specifying a file, you would put another backslash at the end of the path, to separate the path from the 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 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 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 so 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
```

1.7.8 Cleaning your AMSTRAD PPC

We recommend the use of aerosol anti-static foam cleaners to clean both the screen and the plastic case. Under no circumstances should spirit-based cleaners be used.

2. USING MS-DOS COMMANDS

2.1 INTRODUCING MS-DOS

MS-DOS Version 3.3 is an operating system - that is, MS-DOS Version 3.3 is software that is used on the PPC to:

- * **translate commands you type in into instructions that your PPC's processor understands**
- * **manage your program files and data files for you**
- * **run programs for you**

in other words, it controls how the PPC behaves.

Without this software - your PPC can't respond to any instructions you try to give it. With it, the PPC is a powerful general purpose microcomputer.

MS-DOS makes the PPC work in very much the same way as PC-DOS, the operating system used on an IBM PC (which you can also use on the PPC if you want). As a result programs written for the IBM PC can be run under MS-DOS. This automatically makes all the PC-DOS software (of which there is a large amount) available on the PPC.

MS-DOS is controlled by simple instructions which you type in after the prompt, A> (or B>, if you have selected the second drive on a dual floppy disk version of the PPC as your default drive). The instructions that MS-DOS understands are instructions to run some special programs called the MS-DOS commands. These commands let you:

- * **Run programs**
Simple programs; Series of programs
- * **Organise your work**
Create files; Put them into groups; Copy files; Display files; Rename files; Delete files; Protect files; Bring files up to date
- * **Process disks**
Prepare new disks; Make duplicate copies of disks
- * **Handle the PPC's input and output devices**
Change where input is taken from and where output is sent to; Use different input and output devices
- * **Personalise the PPC and make use of the PPC's special features**
Personalise how information is displayed on the screen; set your PPC's internal clock

Some of these commands are built into MS-DOS: these are called Internal commands. The rest are stored in files on your PPC SYSTEM disk: these are called External commands. Internal commands are always available for use: External commands have to be read into the PPC's memory from the disk before the command can be carried out.

If you are a newcomer to computing, we suggest you turn back to the beginning of this manual and read section 1.1 before you go any further. This describes what files, directories, input and output devices, etc. are and how a computer uses these.

Starting to use MS-DOS

Note:

If anything happens while you work through this chapter that isn't explained here, turn to appendix G 'Troubleshooting' and see if you can find out what has gone wrong. If in doubt, consult your dealer.

The steps to using MS-DOS commands are as follows:

* If you haven't switched your PPC on:

a. Start with

- the power selector switch in the EXT.ON position
- the plug out of the A.C. supply socket, if you are powering the unit from the A.C. adaptor or an external AMSTRAD monitor unit. (The power switch on the back of the Monitor should be in its OFF, fully released, position; the signal and power output cables should be connected between the monitor and the PPC.)
- the car battery adaptor unplugged
- no disk in any floppy disk drive.

b. Either:

- i) Plug the Monitor into the A.C. supply, then turn the PPC on by pressing the power switch ON THE MONITOR, or
- ii) Insert the A.C. adaptor jack plug into the socket on the PPC, then plug the adaptor into the A.C. supply, or
- iii) Insert the car battery adaptor jack plug into the socket on the PPC and plug the other end into the cigar lighter socket (most car accessory dealers can supply a suitable socket, if none is fitted to your model of car). You may then need to turn the ignition switch to the 'accessories' position to get power to the cigar lighter socket. Check with the vehicle manual if in any doubt.

CAUTION.

It is inadvisable to have the ignition switch in the normal running position for any length of time with the engine not running, as damage to the ignition system could result.

Running the engine is not recommended when the PPC is being powered from the car battery. The PPC can safely accept the full range of input voltage from a properly working 12 volt car electrical system, but electrical 'noise' may be generated, creating data errors in the PPC.

Of course, you also have the option of switching to your PPC's internal batteries, but note the advice on maximising battery life, given in the introduction to this manual. When you are new to computing (and sometimes even when you are not) it is very easy to become engrossed in what you are doing and spend a very long time at the machine; so use a supply other than the internal batteries whenever you can, especially when using a modem or the disks a lot.

When the power is on, your PPC will display the message 'please wait....', whilst it goes through a built-in system check. If all is well, a message similar to the following will shortly appear on the screen.

```
Amstrad Portable PC640K (v1.8) 09:00 on 10 January 1988
(c) 1987 AMSTRAD PLC

Last used at 18:00 on 09 January 1988

Insert a SYSTEM disk into drive A
Then press any key
```

insert a **SYSTEM** disk into drive **A**

Then press any key

If the batteries are dead, the time and date messages will not appear. Instead, you will be reminded to set the time and date and fit new batteries. You are not obliged to do so; MS-DOS can be loaded and the PPC used, but the time and date used in any labels or messages will be incorrect. If the batteries are not replaced, the time and date will be correctly maintained, once set, until the PPC is switched off.

If you see the following message:

```
Check Keyboard
```

clear everything off the keyboard and check that none of the keys are stuck down. the message will be erased shortly after you clear the problem.

Insert the copy which you made of your PPC SYSTEM DISK into Drive A - the nearest to the keyboard if you have two drives- and push it firmly home.

Press either the space bar or the [←]

You should now see the green, drive indicator, light on the top of the PPC go on and off a few times. When your PPC is ready, it will be showing the following additional messages:

```
A>PATH \;
A>KEYB US 437
A>ECHO OFF
Current date is Tue 1-01-1980
Enter new date (mm-dd-yy):
```

followed by:

```
MS-DOS Version 3.30
With Scroll Lock on:
Right [Alt (Gr)] - LCD text contrast toggle
Right [Ctrl] - LCD/CRT toggle
A>
```

*** If you have already been using your PPC:**

a. Hold down the [Ctrl] and [Alt] keys and press the [Del] key.

b. Insert your PPC SYSTEM disk (preferably, a copy of the supplied disk) into Drive A -the drive nearest the keyboard if you have two. You should now see the green, drive indicator, light on top of the PPC go on and off a few times. When your PPC is ready, it will be showing the following message:

```
Amstrad Portable PC 640K (V1.8) 11:02 on 11 November 1988
(c) 1987 AMSTRAD plc
Last used at 11:02 on 11 November 1988

Insert a SYSTEM disk into drive a
Then press any key
```

then briefly:

```
A>PATH \;
A>KEYB US 437
```

followed by:

```
MS-DOS Version 3.30
With Scroll Lock on:
Right [Alt (Gr)] - LCD text contrast toggle.
Right [Ctrl] - LCD/CRT Toggle.

A>
```

If the batteries are dead, the time and date messages will not appear. Instead, you will be reminded to set the time and date and fit new batteries. You are not obliged to do so; MS-DOS can be loaded and the PPC used, but the time and date used in any labels or messages will be incorrect. If the batteries are not replaced, the time and date will be incorrectly maintained, once set, until the PPC is switched off.

*** You can now start typing MS-DOS commands.**

The A > is the MS-DOS system prompt, which tells you that MS-DOS is ready to receive a command. The oblong that follows the A > is the cursor and its position marks where anything you type will appear on the screen.

Note:

To conserve space in the system's memory MS-DOS does not load the software needed to execute some of the more complex and less frequently used commands into memory, until it is actually needed. These are the External commands, which we referred to earlier in this section; you will need to load your SYSTEM disk again whenever you want to use one of these commands.

Using MS-DOS commands

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 like this to run programs you buy such as word processor programs 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 like this 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:

```
A>COPY FILE.X NEWDATA.PQ
```

The Command Tail, in this case the names of files you want to process

The Command Name, that is, the name of the program you want to run

The System Prompt

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 as you type.

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 or command's **Form** (or **Syntax**). The Form of each MS-DOS command is given in this manual as part of its detailed description in sections 2.4 to 2.7. The Form of the command lines you need to run any commercial programs you buy will be given in the programs' 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 you want to run is stored (that is, in which drive and in which directory) and, similarly, details of where the files you want the program to work on are located. 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. If you receive a message on the monitor to say that the file has not been found, it may be because you have failed to give MS-DOS the correct details of where it is stored.

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 2.4.1.1).

If any of these terms seem strange, do go back and review section 1.1 before continuing.

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 PPC will do exactly the same actions whether your command line is:

```
A>COPY FILE.X NEWDATA.PQ
```

or

```
A>copy file.x newdata.pq
```

or

```
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', below), 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 are more used to the system and are perhaps entering rather long command lines, you will find it helpful to use the command line editor, described in section 2.4.2.1. It enables you to alter parts of what you have typed without erasing the correct parts. It is not difficult to use but, like anything else, it requires a little practise to master properly. You may not want to bother with it for simple commands which you can easily erase and retype in full.

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 drive nearest the keyboard, if you have two). If you have two floppy disk drives, you might change the default drive to Drive B (the rearmost disk drive); the system prompt would then become B>. (How to change the default drive is described in section 2.4.1.2).

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 section 2.7, 'Tailoring your PPC 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 PPC SYSTEM disk)
- an MS-DOS or PC-DOS program;
- a Batch file, to run a sequence of commands (see section 2.4.4).

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 (that is, 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 filename of the file

The command name never includes the filetype of a program or a command 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 those MS-DOS External commands, and other programs, stored **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 (an area of your computer's memory that is used to simulate 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
```

Note:

The location part of the command name is often left out when the 'Form' or 'Syntax' of the command is given, making it up to you to remember to include this information if it is needed. Details of how to include the location in the command line are given in section 2.4.1

More about the Command Tail

The information a program needs in the command tail and the order this is given in 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 sections 2.4 to 2.7 where all the commands are described in detail. The form of the command lines you will need to run the commercial programs you buy will be given in the programs' own user guides.

The Form statement shows you how to give MS-DOS the details of the files you want the programs to process and it tells you how to specify the program options you want to take advantage of. The way it does this is through items like *d: filename, parameter*, and physical-device written 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 you need 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 (that is, 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
```

What you need to replace the placeholders with should be explained either in the notes describing the command line or in a general section covering the conventions used in the manual. There can also be parts of the command line that you can miss out if they are not appropriate. Details of the conventions used in this manual are given in section 2.3 '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 isn't 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 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 2.4.2. 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 appendix G '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 PPC generally seems 'dead', your program file may have been corrupted.**

Reset your PPC (see, section 1.7.3) 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 section 1.3). Then try running your program again. If you get exactly the same failure this time, try running your reserve copy of the program (if you have made one of these). If that fails too, consult your dealer.

Postscript

In principle, you now know enough about MS-DOS to run any well-documented DOS program.

If all you want to use your PPC for is to run commercial programs, you could now go ahead, following the instructions in the programs' own user guides, but we would recommend you also read the following section - 'An overview of MS-DOS commands'. This will give you a broader understanding of what you can do when things go wrong. It will also make getting your programs working that much easier.

2.2. OVERVIEW OF MS-DOS COMMANDS

This section is split into two parts:

- * A summary of each of the reference sections
- * A guided tour through some simple uses of MS-DOS commands

The idea is to give you a feeling for the sort of operations you can use MS-DOS commands for and to get you used to the idea of typing command lines.

When you start using MS-DOS in earnest, we expect you to refer to the relevant section for both background information on the sort of task you want to carry out and for details of the commands you need for a particular task. Each description of a command also includes examples for you to follow.

2.2.1 The tasks you can use MS-DOS for

Running programs

This section 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 monitor 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 PPC is turned on.

Organising your work

This section 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 part of this Introduction 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.

Processing your disks

Processing your disks covers:

*** Preparing new blank disks**

- Dividing up the blank disk into sections so that your PPC 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
- 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 PPC.

Tailoring the AMSTRAD PPC to your needs

When you bought your AMSTRAD PPC, it was set up to be used in a very conventional way - with the keyboard your main means of giving instructions and information (that is, input) to the computer and the LCD screen used by the computer to display (that is, 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 section explains how to tailor your PPC 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 PPC**

- by setting up a personalised system prompt
- by giving your disks names by which you can identify them.

*** Setting your PPC'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 will 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.

2.2.2 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 PPC 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 2.4.1 and 2.5.2. 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 sections 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 [↵] at the end of each command line, to send each command you type to MS-DOS. The more advanced sections of this manual assume that you know you must press [↵] 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 (for example, A>) with the cursor to its right.

Loading MS-DOS

Our guided tour starts by loading MS-DOS. If your PPC is already set up to use MS-DOS commands, join the tour at 'Simple commands'.

Note:

If anything happens while you work through this section that isn't explained here, turn to appendix G 'Troubleshooting' and see if you can find out what has gone wrong. If in doubt, consult your dealer.

* If you haven't switched your PPC on:

a) Start with

- the A.C. plug out of the wall outlet
- the A.C. Adaptor jack plug inserted in the socket on the rear of the PPC
- the power switch on the top of the PPC (visible when the keyboard is opened out) in its EXT.ON position - no disk in any floppy disk drive.

b) Plug the Adaptor into the A.C. wall outlet.

Note:

When you are using one of the alternative sources of supply the procedures are:

External AMSTRAD monitor

- with nothing plugged into jack socket, set PPC power switch to EXT.ON, connect power and signal cables from monitor to PPC; then plug monitor into wall outlet and switch on power switch on rear of monitor.

Car battery adaptor

- set PPC power switch to EXT.ON, insert battery adaptor output jack plug in socket at rear of unit; then plug adaptor into car cigar lighter socket.

Internal battery supply

- with nothing plugged into jack socket, set PPC power switch to BATT.ON. When using this option, remember that the drives (and modem, on PPC 640 models) use quite a lot of power; so try to minimise their use. We strongly recommend using an external supply whilst you are still learning to use the PPC efficiently, and whenever the PPC 640 integral modem is in use.

Your PPC then goes through a built-in system check. If all is well, a message similar to the following will shortly appear on the screen.

```
Amstrad Portable PC 640K (V1.8) 09:00 on 10 January 1988
(c) 1987 AMSTRAD plc
```

```
Last used at 18:00 on 09 January 1988
```

```
Insert a SYSTEM disk into drive A
```

```
Then press any key
```

If the batteries are dead, the time and date messages will not appear. Instead, you will be reminded to set the time and date and fit new batteries. You are not obliged to do so; MS-DOS can be loaded and the PPC used, but the time and date used in any labels or messages will be incorrect. If the batteries are not replaced, the time and date will be correctly maintained, once set, until the PPC is switched off.

If you see the following message:

Check Keyboard

clear everything off the keyboard and check that none of the keys are stuck down. The message will be erased shortly after you clear the problem.

c) Insert your AMSTRAD PPC SYSTEM disk (preferably, a copy of the supplied disk) in Drive A - the drive nearest the keyboard, if you have two.

d) Press either the space bar or [←]

You should now see the green drive indicator light, on top of the PPC, go on and off a few times. When your PPC is ready, it will be showing the following additional messages:

```
A>PATH \;  
A>KEYB US 437  
A>ECHO OFF  
Current date is Tue 1-01-1980  
Enter new date (mm-dd-yy):
```

and, after you have entered the date and time;

```
MS-DOS Version 3.30  
With Scroll Lock on:  
Right [Alt (Gr)] - LCD text contrast toggle.  
Right [Ctrl] - LCD/CRT Toggle.  
A>
```

*** If you have already been using your PPC:**

a) Hold down the [Ctrl] and [Alt] keys and press the [Del] key.

b) Insert your AMSTRAD PPC SYSTEM disk (preferably, a copy of the supplied disk) into Drive A - the drive nearest the keyboard, if you have two.

You should now see the green drive indicator light, on top of the PPC, go on and off a few times. When your PPC is ready, it will be showing the following:

```
Amstrad Portable PC 640K (V1.8) 11:02 on 11 November 1988  
(c) 1987 AMSTRAD plc  
Last used at 11:02 on 11 November 1988  
  
Insert a SYSTEM disk into drive a  
Then press any key
```

then, briefly:

```
A>PATH \;  
A>KEYB US 437
```

followed by:

```
MS-DOS Version 3.30  
With Scroll Lock on:  
Right [Alt (Gr)] - LCD text contrast toggle.  
Right [Ctrl] - LCD/CRT Toggle.  
A>
```

If the batteries are dead, the time and date messages will not appear. Instead, you will be reminded to set the time and date and fit new batteries. You are not obliged to do so; MS-DOS can be loaded and the PPC used, but the time and date used in any labels or messages will be incorrect. If the batteries are not replaced, the time and date will be correctly maintained, once set, until the PPC is switched off.

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 PPC, the default directory is the main or 'Root' directory on Drive A.

So we will start with some command lines that will work with just this directory and we will use Internal rather than External commands because we don't have to take any special steps to ensure that these are available for use, even when you haven't got the SYSTEM disk in the drive. (You will, however, have to have a copy the SYSTEM disk in a drive before you can use any of the MS-DOS External commands.)

Often, the first thing you will want to do is to find out what files are held in this directory. (The reference section that describes this is section 2.4.1.) 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. Up on your screen will come something like this:

```
A>dir
Volume in drive A has no label
directory of A:\

APPEND   EXE   5794   7-24-87  12:00a
ATTRIB   EXE  10656   7-24-87  12:00a
RPED     EXE   4644  10-26-87  10:03a
REPLACE  EXE  13234   7-24-87  12:00a
COMMAND  COM  25276   7-24-87  12:00a
CHKDSK   COM   9819   7-24-87  12:00a
COMP     COM   4183   7-24-87  12:00a
DISKCOMP COM   5848   7-24-87  12:00a
DISKCOPY COM   6264   7-24-87  12:00a
          9 File(s)  459776 bytes free
```

This gives you the names of all the files in the default directory.

DIR 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 particular file is in the default directory. If the file isn't 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 EXE, you would use the template *.EXE and the command:

```
DIR *.EXE [←]
```

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 *.EXE to tell MS-DOS to list all files with the filetype EXE because you can just type DIR *.EXE

If you wanted to list all the files with the filetype EXE whose filenames started DISK, you might use the template D*.EXE and the command:

```
DIR D*.EXE [←]
```

This will, however, give you all the files that match this template, that is, 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 [↵]
or
DIR /P /W [↵]
```

The order in which you list the extra options doesn't usually 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 B (a second floppy disk) without first changing the default drive to Drive B. If you type:

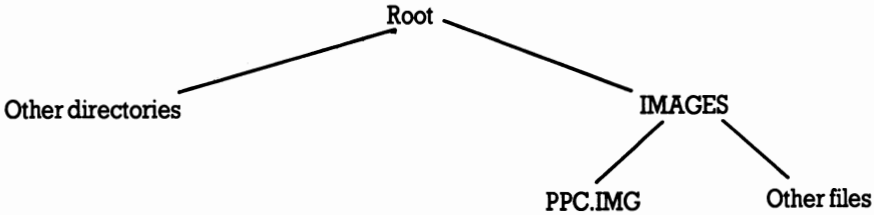
```
DIR B:COMMAND.COM [↵]
```

the AMSTRAD PPC 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 AMSTRAD PPC gives details of a file called COMMAND.COM in the current directory on Drive B (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 PPC.IMG and it is stored in a directory called IMAGES that branches off the Root directory on Drive A.



The path to the file is therefore Root → IMAGES → PPC.IMG, which you write as \IMAGES\PPC.IMG (the first backslash telling MS-DOS to start the path from the Root directory). Drive A is the default drive, so your DIR command line becomes:

```
DIR \IMAGES\PPC.IMG [↵]
```

If Drive A hadn't been the default drive, the command line would have been:

```
DIR A:\IMAGES\PPC.IMG [↵]
```

Postscript

This brief tour should have given you the flavour of how you use MS-DOS commands to carry out tasks for you. In the rest of this part of the manual, we describe the tasks and the commands you will need to use in detail.

2.3 CONVENTIONS

The following reference sections 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* (that is, 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 *filename* 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 the table below.

In addition, optional parts of the command line are surrounded by slanting square brackets; these parts may be left out altogether.

For example:

```
DATE [mm-dd-yy]
```

means that depending on the precise details of what you want to do, you either type:

```
DATE mm-dd-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 2.4.2.)

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 be shown as a ^ character.

Standard placeholders

<i>command</i>	a complete command line, here incorporated in another command line
<i>d</i>	the drive holding the file you want to use represented as a letter. Often followed by a colon: for example, <i>d:</i> can be replaced by <i>A:</i> , <i>B:</i> , <i>C:</i> or ...
<i>destination</i>	the name of the file you want the copy to be stored in, written as <i>filename.filetype</i> (see below), or the name of a logical device you want the contents of a file to be sent to. (See section 2.5.2.)
<i>device</i>	the name MS-DOS recognises for an input or output device attached to your PPC. (See section 2.7.1.)
<i>filename</i>	the first part of a file's name. For example, if the file's name is MYFILE.TXT, then the <i>filename</i> is MYFILE
<i>filetype</i>	the second part of a file's name. For example, if the file's name is MYFILE.TXT, then the <i>filetype</i> is TXT
<i>n</i>	a number. Details of the range of numbers that can be used will be given in the description of the command

<i>path</i>	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 2.4.2.)
<i>source</i>	the name of a file you want to copy written as <i>filename.filetype</i> (see above) or of a logical device from which you want to take information and store it in a file. (See section 2.5.2.)
<i>source-drive</i>	the drive holding the disk you want to copy, written as <i>d:</i> (see above).
<i>string</i>	a string of characters or a parameter that has been set to a string of characters
<i>target-drive</i>	the drive holding the disk you want to store the copy on, written as <i>d:</i> (see above).

Structure of the reference sections:

The reference sections each deal with a different aspect of using your computer:

- * **running programs**
- * **organising your work**
- * **processing disks**
- * **tailoring your PC to your needs**

To make it easier for you to find the information and the help you need, each section has been given the same structure:

- 1 A general introduction to the section, giving details of the sections it contains**
- 2 Sections dealing with specific aspects of the tasks, each major section starting on a new page**
- 3 Details of all the commands involved in this area of work, listed in alphabetical order**

The descriptive sections describe how to set about achieving the result you require and tell you about the commands you will need to use. The details of each command include full information on how the command should be used and on the command lines you will need, alongside examples showing the command's use in practice.

In the early stages, you will need to look first at the descriptive section to discover which command to use and then at the details of the command to find out the command line you need to type. Most of the commands it refers to will be detailed at the end of the same section, but occasionally you will be referred to another section for full details of the command.

Later, you will probably rely most on the detailed command information, with only occasional references back to the descriptive introductions. To identify the part of the description you need to help you use a particular command, look in the index for the page number.

2.4. 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
- a 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 stored on a disk in one of your drives and MS-DOS is told where to look for them.)

In section 2.4 of this manual, we describe the essential steps in running a program on your PPC, using the 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 section, we look at the additional facilities that MS-DOS provides.

Section 2.4.1 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
- finally, typing in the command line that starts the program running.

Section 2.4.2 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 they need at the keyboard. They are often also set up to send any information they generate to the monitor screen. Section 2.4.3 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 other programs and process it.

Section 2.4.4 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 PPC to run whenever it is turned on.

When you buy programs for your PPC to run under the MS-DOS operating system, remember you need to choose ones that are:

- * **stored on 3¹/₂ inch, 720KB, floppy disks**
- * **designed to run under either the MS-DOS or the PC-DOS operating system**

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 PPC. You can readily find this out by using the VER command.

Programs supplied with your PPC 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 appendix A.

To understand the commands covered in this section, it is important to know what a directory is. If you aren't certain you know, read section 1.1 'Fundamentals of computing' before you use the commands.

2.4.1 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 a 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**
- * **and finally, running the program**

Notes:

- i) If the program you want to use wasn't specifically intended for use on the PPC, you may well 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 PPC. How to go about this is described in Appendix A.
- ii) Always remember to check out 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, for use solely to make further copies as and when these are needed.
- iii) There is advice in Appendix A 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).

2.4.1.1 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 the PPC. 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?

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 it which directory you want information from
- specifying the names of the files within the directory that you want information about.

You specify a directory by giving the drive it is on and the path to the directory. (How to do this is described in Section 2.4.1.2). If you don't specify a directory, your PPC 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 1.7.5).) If you don't 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 details of five files on each line. If you don't specify a 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 PPC.

2.4.1.2 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:

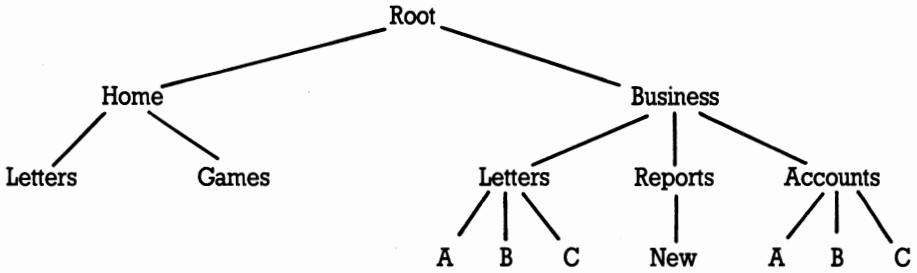
- to use a particular file from a different directory.
- to change the current directory on the drive, for example if the command or program you want to use can only process files in the current directory

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.

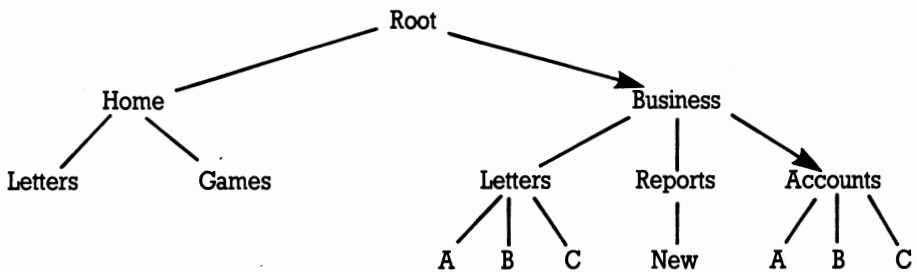
The starting point for the path is always: either the root directory on the drive or the directory you are currently working with on the drive. You can use the path from the root directory regardless of which directory you are currently working with.

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

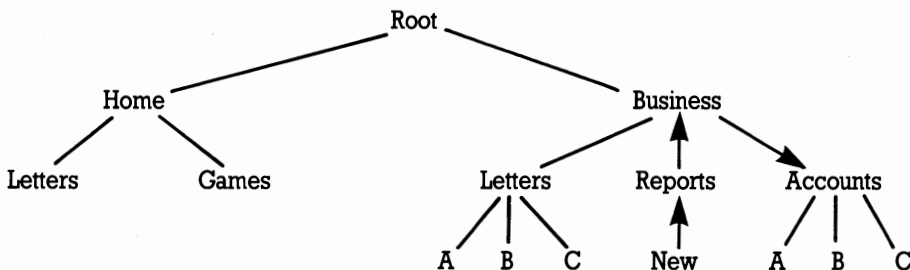
When you come to use this path, for example to specify a file in the directory, you precede it by a backslash (to tell the operating system that the path starts at the Root directory). If you are specifying a file, you would put another backslash at the end of the path, to separate the path from the 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 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 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'

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.**

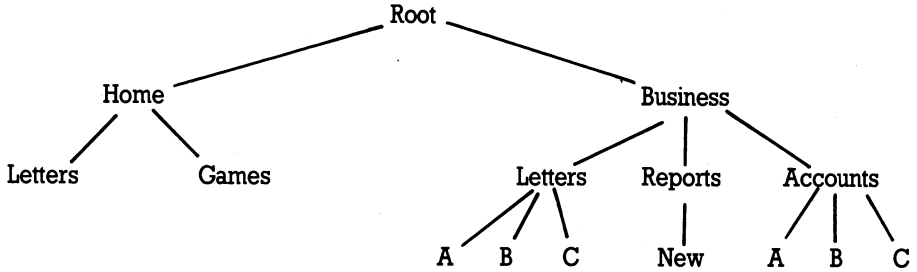
For example, to specify a file called NEW.A in the Accounts directory, you would write:

```
..\..\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 don't 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.

To change which directory is automatically used on any drive or to see which the current directory is, use the `CHDIR` ('Change directory') command.

Using the `CHDIR` command on the default drive changes the default directory but using it on any other drive doesn't. To make the current directory on another drive the default directory, you have to change 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 B (a second floppy disk) you should type:

```
B: [↵]
```

Having more than one current directory on a 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.

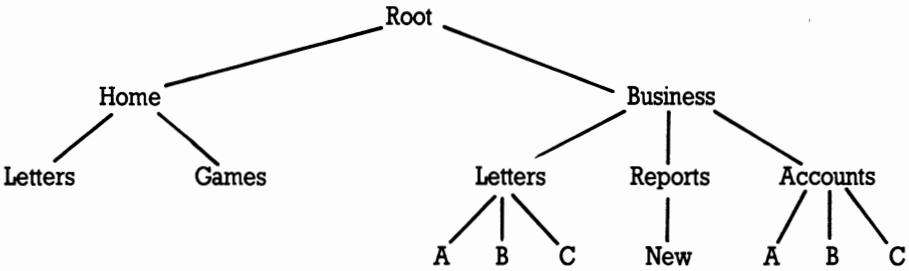
There are two possible sets of circumstances:

- running a suite of programs when the program files and MS-DOS commands 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 2.4.2). The second is handled by treating the directories as if they were on separate drives.

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 PPC doesn't already have a meaning for, for example Drive N, Drive O or Drive P. (The PPC initially has meanings for Drives A...E.)

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, say, 'Accounts' the current directory on the real drive and assign 'Reports' to your 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, that is:

```
N:REPORT.85
```

The directory assigned to the imaginary drive acts as the root directory on that drive. If this directory has any 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.

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.

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.

2.4.1.3 Running two-disk programs on a single-drive PPC

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 PPC, 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 (for example, 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.

2.4.1.4 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.

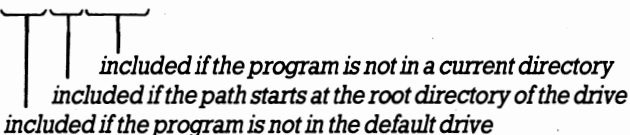
(A few programs may additionally require certain parameters to be set up in advance, which are used to specify the Environment the program works in. For example, the program may work differently depending on the current size of your PPC's RAM disk. If this is the case, full details should be given in 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 carriage-return 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 this manual in section 2.3.

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 section 2.4.2.4). You will need to precede the command name with:

`[d:][\][path]`


included if the program is not in the 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 [↵].

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 appendix G. This covers troubleshooting a wide range of problems. If this doesn't solve your problems for you, seek the advice of your dealer.

Note:

There are two other MS-DOS commands you may find helpful when you are running application programs:

- | | |
|---------------------|--|
| <code>CLS</code> | which clears the screen and puts a fresh system prompt in the top lefthand corner |
| <code>VERIFY</code> | which 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 |

2.4.2 SHORTCUTS AT THE KEYBOARD

2.4.2.1 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 [↵], you can always use the [←] 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 'forwards' 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, and
- every time you type fresh characters without telling MS-DOS that you want to insert these.

It also moves 'backwards' if you delete any characters from the line.

The keys you want are as follows:

- | | |
|---|---|
| [F1] | Add the next character from the stored line. |
| [F2] character (that is, 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 (that is, press [F4] and then a character key) | Add the given character and everything in the stored command line to the right of this character. |
| [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 stored 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 to MS-DOS for processing. It also replaces the stored command line with this new line. |

For example, suppose that your last command was:

```
COPY MYFILE B:\DIR1\FILE
```

and you wanted the next command to be:

```
COPY MYFILE C:\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, that is:

```
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 l, 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
```

2.4.2.2 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] [C]** or by holding down the **[Ctrl]** key and pressing the **[Break]** key.

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 2.5.2) 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.

2.4.2.3 Keeping a record of the work you do

If you have a printer attached to your PPC, you can use this to record everything that appears on the screen

- that is, 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, you enter **[Ctrl] [P]**. A second **[Ctrl] [P]** stops this information being output on the printer.

However, don't 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, don't.

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 PPC by using the **GRAPHICS** command for your printer.

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 screen display.

Provided your PPC has been properly set up for your printer, a picture of your screen will be printed in a few minutes.

DON'T PRESS THE [PrtSc] KEY IF THERE ISN'T A PRINTER ATTACHED TO YOUR PPC

2.4.2.4 Searching for files

A tidy approach to using your PPC 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 directories 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 2.4.4.)

You will find this particularly helpful in using applications that make use of development software (for example, a programming language such as BASIC) as well as its own programs or when you need to use MS-DOS external commands alongside your applications 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.

If you type in a command line that doesn't 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 PPC, which loses all the details of the search path).

The PATH command is also used to display the directories that are searched at present.

Note:

You must keep the list of directories to be searched up to date. If you delete a directory on the search path (see section 2.5.1), 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

If you type in a command line that doesn't specify where the files you want to process are located, MS-DOS will always look for 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 the list) - until, that is, you use the APPEND command again (or you reset your PPC, 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.

2.4.3 REDIRECTING INPUT AND OUTPUT

Programs are usually designed to take their standard input from the keyboard and to send their standard output to the monitor. Under MS-DOS, however, programs 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 2.7.1.

2.4.3.1 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:

<i><device-name</i>	if the input is to be taken from a different input device
<i><[d :][\][path \]filename . filetype</i>	if the input is to be taken from a file table

A list of device names is given in appendix D. For example, if the input for the program MYPROG is to be taken from the serial interface, you might have the command line:

```
A>MYPROG <AUX  
or A>MYPROG <COM1
```

If, however, it is to be taken from the file RUNDATA.1 on Drive B, you might have the command line:

```
A>MYPROG <B: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.

2.4.3.2 Redirecting the standard output

You 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:

> <i>device-name</i>	if the output is to be sent to a different input device
>[<i>d:</i>][\][<i>path</i> \] <i>filename . filetype</i>	if the output is to be sent to a file (replacing any existing file of that name)
>>[<i>d:</i>][\][<i>path</i> \] <i>filename . filetype</i>	if the output is to be appended to an existing file

A list of device names is given in Appendix IV.

For example, if the output from the program MYPROG is to be sent to the printer, you might have the command line:

```
A>MYPROG >PRN  
or A>MYPROG >LPT1
```

If, however, it is to be appended to the file RUNDATA.1 on Drive B, you might have the command line:

```
A>MYPROG >>B:RUNDATA.1
```

2.4.3.3 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:

```
A>DIR B: | FILTER >STORE
```

where `DIR B:` 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**

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 B: | SORT | MORE
```

Note:

You cannot make use of piping within a Batch process.

2.4.3.4 Commands used in running programs

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 alongside 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:

- i) Each path must start from a root directory.
- ii) 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.
- iii) 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 4.2.4) 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...]*

	<i>Searched drive</i>
	<i>Requested drive</i>

* To clear one assignment

Form ASSIGN *d=d*

	<i>Requested drive</i>
	<i>Requested drive</i>

* To clear all current assignments

Form ASSIGN

Note:

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 A. The command line you need is:

ASSIGN B=A

(assuming that the external command ASSIGN is stored either in the default directory or in a directory that MS-DOS automatically searches - see Section 2.4.2.4)

- * You have assigned both Drive A and Drive B to Drive B, but now you want to have requests for Drive A going to Drive A. The command line you need is:

ASSIGN A=A

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 isn't 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

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, don't 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

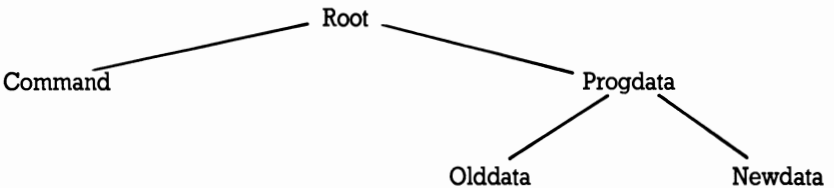
Form CHDIR [d:]

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 backslash is used 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:

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 B is the current directory. The command line to type is:*

```
CD B:
```

and the response might be:

```
B:\DIR1
```

CLS

Internal command

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 ('green' characters on a 'black' background), it will go back to its normal mode of operation ('black' characters on a 'green' background) after the screen is cleared. Highlighting is also removed.

Example:

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

APPEND   EXE     5794   7-24-87  12:00a
ATTRIB   EXE    10656   7-24-87  12:00a
RPED     EXE     4644   10-26-87 10:03a
REPLACE  EXE    13234   7-24-87  12:00a
COMMAND  COM    25276   7-24-87  12:00a
CHKDSK   COM     9819   7-24-87  12:00a
COMP     COM     4183   7-24-87  12:00a
DISKCOMP COM     5848   7-24-87  12:00a
DISKCOPY COM     6264   7-24-87  12:00a
          9 File(s)  459776 bytes free

A>cls
```

Press [↵]

```
A>
```

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. Which style of display is used depends on 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 - that is, filename, filetype, size and date when last changed - all the files in the default directory (that is, 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 B, use the command line:

```
DIR B: /W
```

- * To display full details - that is, filename, filetype, size and date when last changed - of the files in the Root directory on Drive B, use the command line:

```
DIR B:\ /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 wildcard character * is used 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.

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 (for example, *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 2.4.2.4)

If you want just the word *computer*, you might use the command line:

```
FIND " computer " MYFILE.TXT
```

though this wouldn't pick out *computer.*, *computer*, etc.

- * You want to display and number all the lines in *MYFILE.TXT* that contain the phrase "Best Fit", complete with quote marks. The command line to use is:

```
FIND/N ""BestFit"" MYFILE.TXT
```

Note:

The complete expression, including the quotation marks, is required; therefore two sets of quotation marks must be used in the command line.

- * You want display full details of all the files in the current directory that were last changed on a particular date, for example September 26, 1986. To do this, you want to use the instruction *DIR* to list the directory but to filter this list before it is displayed on the screen by using the *FIND* command to filter out just those lines containing the date 09-26-86. The *FIND* instruction you want is *FIND "09-26-86"*, giving you a complete command line of:

```
DIRIFIND "09-26-86"
```

GRAPHICS

External command

GRAPHICS *printer-type*[/R][/B]

Prepare your PPC to print graphics screen displays

Note:

When operating the PPC with a monochrome display (MD) no graphics are available, so the GRAPHICS command is redundant in that case.

The GRAPHICS command prepares your PPC so that graphics displayed on the screen can be correctly reproduced on the printer attached to your PPC. This can be either a colour or a monochrome printer.

Through this command, you can chose whether the screen display is printed as 'light' text on a 'dark' background or as 'dark' text on a 'light' background and, if you have a colour printer, whether the background is all coloured in the background colour. These features are selected through the /R and /B options.

Form GRAPHICS [*printer-type*][/R][/B]

Printer types	COLOR1	IBM Personal Computer Color Printer (or compatible) with a black ribbon
	COLOR4	IBM Personal Computer Color Printer (or compatible) with an RGB (red, green, blue and black) ribbon
	COLOR8	IBM Personal Computer Color Printer (or compatible) with a CMY (cyan, magenta, yellow and black) ribbon
	COMPACT	IBM Personal Computer Compact Printer (or compatible)
	GRAPHICS (or nothing)	IBM Personal Computer Graphics Printer (or compatible, for example the AMSTRAD DMP3000)
	COLORJET	IBM Color Jet Printer (or compatible)

Options

/R	Print foreground light, background dark, as seen on the monitor. If this option isn't selected, then the foreground is printed dark and the background light.
/B (printer-types COLOR4 and COLOR8 only)	Print the background colour. If this option isn't selected, the background colour isn't printed.

- | | |
|-----|---|
| / F | Turn the screen dump through 90degrees. |
| / C | Centre the screen dump on the page. |

Notes:

- i) Using the GRAPHICS command reduces the amount of RAM available for processing other commands and for running programs. This memory space is not made available again until your PPC has been switched off or reset.
- ii) 320 x 200 pixel screens are printed with the top of the screen towards the top of the sheet of paper. 640 x 200 pixel screens are printed with the top of the screen along the lefthand edge of the paper.

Examples: _____

- * *To set up your PPC to work with the Amstrad DMP3000 printer (or some other printer that is compatible with the IBM Personal Computer Graphics Printer), use the command line:*

GRAPHICS

(assuming that the external command GRAPHICS is stored either in the default directory or in a directory that MS-DOS automatically searches - see section 2.4.2.4)

Once this instruction has been used, you can then print out a picture of your screen at any time by pressing [PrtSc] until you reset your PPC.

Note:

The foreground will be printed in black and the background will be white.

- * *To set up your PPC to work with an IBM Personal Computer Color Printer (or compatible printer) with a black ribbon so that it prints the foreground in white against a black background, use the command line:*

GRAPHICS COLOR1/R

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 the drive is joined to must be empty and it must be in the root directory of the second drive. If the directory you specify in the JOIN statement doesn't 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

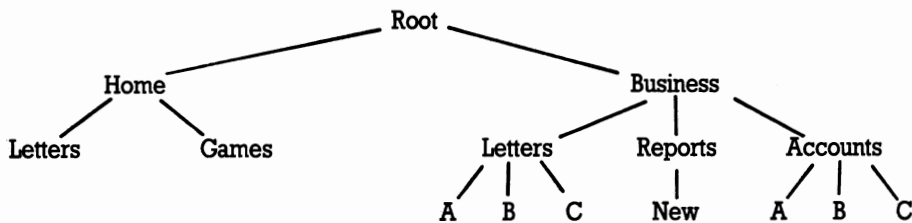
Form `JOIN`

Notes:

- i) After you have joined a drive to a directory, don't try to use the joined drive in the normal way. MS-DOS now regards this drive as an 'Invalid drive'.
- ii) If the directory isn't empty when you try to make the join, the message Directory not empty will be displayed.
- iii) You cannot break the join between a drive and a directory if the file holding the external command JOIN is in the drive.

Examples:

Suppose the directory structure on the disk in Drive A is like this:



- * To put the disk on Drive B into this directory structure, you would type a command line of the form:

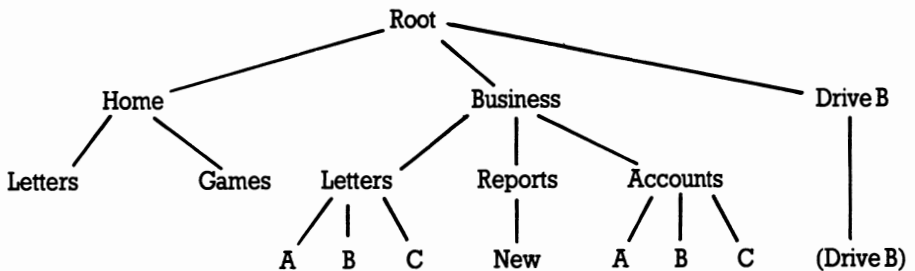
```
JOIN B: A:\path
```

(assuming that the external command JOIN is stored either in the default directory or in a directory that MS-DOS automatically searches - see Section 2.2.4)

- * To make a specially created directory called DriveB represent the drive, use the command line:

```
JOIN B: A:\DRIVEB
```

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 B part of the Home directory and so type the command line:

```
JOIN B: A:\HOME
```

This will be rejected because the directory must be empty. The Home directory has at least two subdirectories in it (that is, Letters and Games)

- * Knowing that the directory called New was empty, you might type the command line:

```
JOIN B: 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 B and the directory DriveB, use the command line:

```
JOIN B: /D
```

MORE

External filter

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 [↵], 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.RAWIMORE
```

(assuming that the external filter MORE is stored either in the default directory or in a directory that MS-DOS automatically searches - see section 2.4.2.4)

Press [↵] to see the next page of output.

PATH

Internal command

PATH [d:]\path[;[d:]\path...]

Define a search path.

Normally, if you don't say which directory a file is in, MS-DOS tries to find it in the default directory. PATH sets up a sequence of directories that 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:

- i) Each path must start from a root directory.
- ii) 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.
- iii) 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:

- * *Suppose you have stored all the MS-DOS external commands in a directory called \COMMANDS on Drive B, but have no search path set up at present. To get MS-DOS to search for the external commands as if they were always in the default directory, use the command line:*

```
PATH B:\COMMANDS
```

- * *Alternatively, if the current search path is:*

```
A:\MYPROGS;A:\MIRROR
```

and you want to extend this list to include B: COMMANDS. The command line to use is:

```
PATH A:\MYPROGS;A:\MIRROR;B:\COMMANDS
```

- * *Now suppose you decide to 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:\MYPROGS;A:\MIRROR;B:\COMMANDS

the command line you need to type is:

PATH A:\MYPROGS;A:\MIRROR

- * *If you decide to clear the path, thus making the default directory the only directory to be searched, use the command line:*

PATH ;

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 2.4.4).

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 parameter 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 parameter from the environment**

Form SET *parameter* =

*** To list the current environment strings**

Form SET

Notes:

- i) 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.
- ii) The list of current environment parameters and strings will contain a number that you won't know have been set. These are likely to be used by MS-DOS and shouldn't 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 ON or OFF. To set DISPLAY to ON, type the command line:

```
SET DISPLAY=ON
```

Later, you can set it to OFF by typing:

```
SET DISPLAY=OFF
```

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=A:\COMMAND.COM  
DISPLAY=on  
  
A>
```

Note:

There are some parameters included which you haven't used a SET command to set. These have been set by other MS-DOS commands or by the operating system itself.

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 character 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, that is, sort from Z 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 2.4.2.4)

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 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 isn't 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*
 └──┬──────────┘
 The imaginary drive The path that is substituted

* To end the substitution

Form SUBST *d*: /D
 |
 The imaginary drive

* To display current substitution

Form SUBST

Notes:

- i) You can form but cannot break a substitution that involves the directory holding the external command SUBST. You may need to reset your PPC in order to break the connection with the imaginary drive.
- ii) 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 the 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 imaginary 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 2.4.2.4)

Later, when you want to break the association between the directory and Drive E, use the command line:

```
SUBST E: /D
```

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 doesn't 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 PPC will display whether verification is currently turned on or off.

* To turn verification on

Form VERIFY ON

* To turn verification off

Form VERIFY OFF

* To show current state

Form VERIFY

Notes:

- i) While verification is turned on, you will only get a message on the screen when a disk error has been spotted.
- ii) Do not assume that verification is turned on because you haven't entered a VERIFY OFF command. Verification can be turned off from within a program.

2.4.4 SETTING UP A SEQUENCE OF COMMANDS

As you use your PPC more, you are likely to find yourself 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.

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.

2.4.4.1 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 A:\WS
FUNCTION
WS.PFK WS
```

you create a file with these three lines as the lines of the file. (Use RPED for this).

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 A:\WS
FUNCTION
WS.PFK WS
```

Notice that you type RUNWS and not RUNWS.BAT.

The command lines will be displayed on the screen one by one as your PPC executes them.

Note:

If a command line you want to include in a Batch file contains the character % (other than in a dummy parameter - see Section 2.4.4.3), type this as %%

2.4.4.2 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 PPC'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 these is described below, with full details of the Subcommands given at the end of the section.

Repeated or conditional commands within Batch files

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 PPC's progress on the screen

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 PPC 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

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 PPC will stop after the `PAUSE` subcommand. You can then change disks or whatever. When you are ready for your PPC 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.

2.4.4.3 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 2.4.1.4). 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 ON or OFF, you would include the parameter %DISPLAY% in an IF subcommand in the Batch file like this:

```
IF "%DISPLAY%"=="ON" 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 %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
```

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 at the end of this section.

2.4.4.4 Interrupting a Batch process

You interrupt a Batch file while it is 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 of 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.

2.4.4.5 Obeying a Batch file automatically

```
> AUTOEXEC.BAT >
```

`MS-DOS` will obey a Batch file automatically when you turn your `PPC` on or reset it, if this file is called `AUTOEXEC.BAT` and it is stored in the root directory of your Startup disk (that is, the disk you use to load `MS-DOS` from).

Immediately after the operating system software has been loaded into your `PPC`'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 `PPC` 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.

The `AMSTRAD PPC` is supplied with an `AUTOEXEC.BAT` file on the `SYSTEM` disk that:

- sets up your keyboard as a US keyboard (see Appendix E)
- can be modified, using `RPED`, to automatically load the program which you use most, a word processor or `MIRROR II` perhaps.

2.4.4.6 Batch command and sub-commands

BATCH

Internal commands

[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 in which they appear in the Batch file. A Batch file can contain commands to execute other batch files, which return control to the next command in the original Batch file when they terminate.

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-1 [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:

- i) The Batch file must have the filetype BAT. The BAT filetype is not included in the command line.
- ii) The Batch file can contain the names used to represent environment strings as parameters (see section 2.4.1). 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.
- iii) 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.
- iv) 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 original 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
IF NOT	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

Note:

If you don't want to use ECHO to stop all the commands being displayed on the screen, but you don't want each of the commands in the Batch file to be echoed, you can suppress the echoing of any line in a Batch file by putting an @ as the first character. (You can also use it before the ECHO command, to prevent that from appearing).

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 PPC
 - makes a directory called *123FILES* the current directory on the default drive
 - runs Lotus 1-2-3, would contain the lines:

```
MODE LPT1:=COM1:
CD \123FILES
LOTUS
```

Set this simple Batch file up either by using the *AMSTRAD* PPC text editor *RPED* or with a *COPY* command (see section 2.5.2.1).

- * 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
```

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:

- i) 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.
- ii) 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).
- iii) When ECHO is ON you can suppress the display of any line in your Batch file by putting an @ character before it.

Examples

- * You don't 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
```

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 PPC 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:

- i) 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.
- ii) *command* cannot be another FOR subcommand.
- iii) The list of values can include a group of filenames represented by a file name template, created using the wildcards * and ? (see section 1.7.5). 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 by %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 B, include the command line:

```
FOR %A IN (OLDDATA.1 NEWFILE.2) DO COPY %A B:
```

This is exactly equivalent to including the lines:

```
COPY OLDDATA.1 B: COPY NEWFILE.2 B:
```

To make the Batch file copy to Drive B first 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 B:
```

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 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
```

GOTO

Batch subcommand

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 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:

- i) 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.
- ii) 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 isn't found, the rest of the Batch file is abandoned.
- iii) 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 at the appropriate place in your Batch file the command line:

```
GOTO Stage 2
```

IF

Batch subcommand

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 environment 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:

- i) The total command line may not exceed 127 characters (including spaces).
- ii) Of the MS-DOS commands, only BACKUP, FORMAT, REPLACE, RESTORE and XCOPY return an exit code. However, your own programs may return exit codes.
- iii) The strings that are compared may not include tab, space or equals characters.
- iv) The two strings will not be regarded as the same if they have the same characters but don't match in which letters are upper and lower case.

Examples:

To copy the file *OLDDATA.1* on the default drive to Drive B 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 B:
```

To copy the file *OLDDATA.1* to Drive B only if the previous command has been successful (an exit code of 0), use the command line:

```
IF NOT ERRORLEVEL 1 COPY OLDDATA.1 B:
```

To check that the file *NEWFILE.2* exists before copying it to Drive B, use the command line:

```
IF EXIST NEWFILE.2 COPY NEWFILE.2 B:
```

To copy *NEWFILE.2* to Drive B 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 B:
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
```

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.

If the PAUSE command includes a remark, this is displayed as processing stops. You can use this, for example, to prompt for a specific disk or to remind you to check that the printer is turned on.

Form PAUSE [*remark*]

Notes:

- i) 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]**.
- ii) 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.
- iii) 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 your 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:

```
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.

REM

Batch subcommand

REM [*remark*]

Display remark

REM marks the point in the Batch file at 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:

- i) 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.
- ii) The remark isn't 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.

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 won't 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.

SHIFT

Batch subcommand

SHIFT

Shift the dummy parameters

SHIFT is used to change what value each dummy parameter in the Batch file has. 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 only use 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 in the Examples below.

Form SHIFT

Note:

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 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 (12) 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 make the Batch file just the commands and to run it with each of the files in turn, that is, 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 File 1, the second time it would be File 2 and the third time it would be File 3. At the start of the next loop, %1 would be blank and so the IF command will direct processing to the label :END

2.5. ORGANISING YOUR WORK

This section describes how to make and keep your files readily available for use by:

* **Putting your files into groups**

As you increase the number of a 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 2.5.1 describes how to

- Display the current pattern of groups on a disk
- 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 section 2.5.2 help you put this right by:

- Creating new files
- Making copies of existing files
- Displaying existing files
- Erasing files you no longer need
- Changing what files are called

* **Protecting important files**

Files are readily erased or overwritten. section 2.5.3 describes how to protect your most valuable files - programs you have bought, last year's accounts, etc. - against being erased or overwritten accidentally.

2.5.1 PUTTING FILES INTO GROUPS

Files on DOS media are organised into directories which are themselves organised into a 'tree-like' structure. As you use your PPC more, you will want to change the structure of the directories by:

- * **adding new directories**
- * **removing directories you no longer need**

This section describes how to make these changes to the structure.

Note:

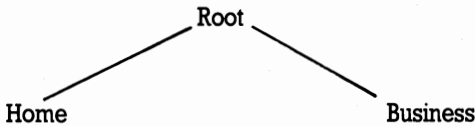
It isn't possible to change the name of a directory.

Before you make any changes you will need a clear idea of how the directories are already organised. For this, you need to know both how directories are organised in general and how to build up a picture of the tree by working steadily through the directories, moving from each directory to an adjacent one. How directories are organised is described in section 1.1 and moving from one directory to another by using the CHDIR command is described in section 2.4.1. Do read these sections if you haven't already done so.

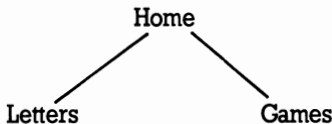
2.5.1.1 Displaying the current pattern of directories

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 2.4.1) 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.

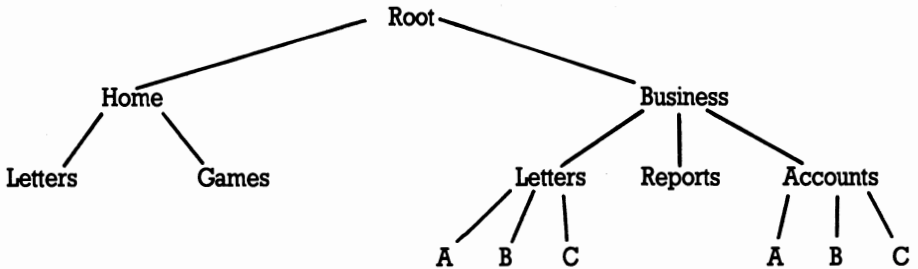
For example, if the Root directory has the subdirectories 'Home' and 'Business', the top part of the tree must be:



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' and so did the 'Accounts' directory. If it turned out that neither of the 'A', 'B' and 'C' directories nor the 'Reports' had any subdirectory, your final directory tree would look like this:



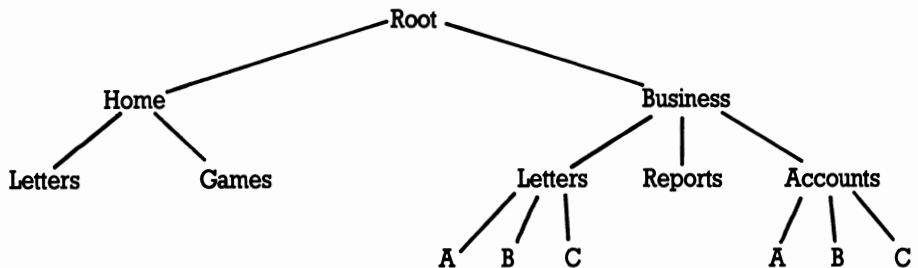
Working through the directories in this way is not the only method of discovering what directories you have and how these fit together. You can also use the TREE command, details of which are given at the end of this section.

The TREE command gives you a list of all the directories on a disk. This list contains details of the path from the Root directory to each directory listed and the names of 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.

2.5.1.2 Adding a new directory

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

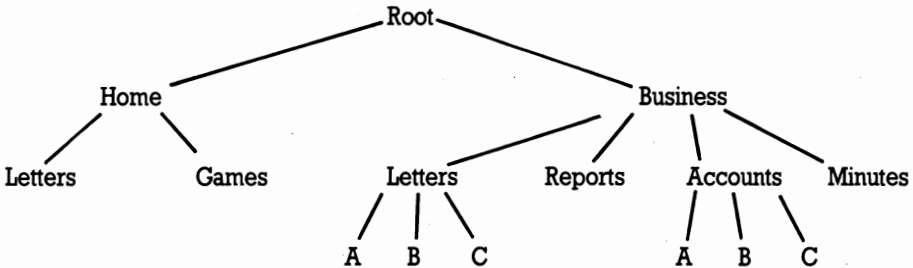
- * **either from the root directory**
- * **or from the current directory**

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 (see section 1.7.6) 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` and the command line to create the directory would be:

```
MKDIR \BUSINESS\MINUTES
```

When a directory has just been created, it doesn't 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 section 2.5.2) 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 details of the route to the root directory and to the parent directory, respectively. All directories other than the root directory have these entries.

2.5.1.3 Removing a directory

A directory is removed from the directory structure 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 2.5.2.3). But do take care to clear only the directory you mean to clear!

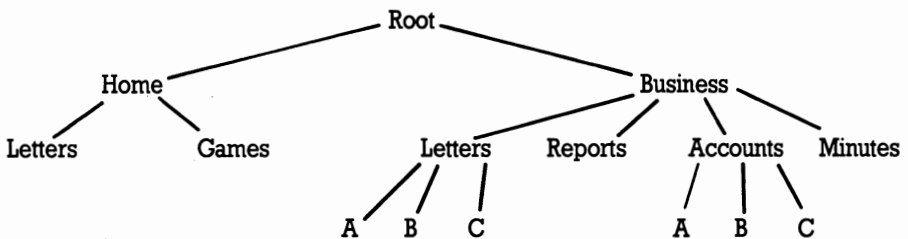
Note:

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 2.4.1)**

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.

2.5.2 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, store this in the Directory B and then delete the original file from Directory A.

This section looks at the commands needed to find out about the 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 AMSTRAD PPC's text editor RPED (see section 1.6.8). Alternatively, you might use MS-DOS's text editor EDLIN described in the reference part of this section. 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 2.5.2.1, which follows.

To use the commands described here effectively, you need to understand about DOS directories. If you aren't certain you do, read section 1.1 of this manual before you read any further. You will also need to know what paths are. These are described in section 2.4.1.2

2.5.2.1 'Making copies of files'

You make copies of files to:

- * **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 (erasing the original file once the copy has been made and checked)**
- * **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, details of which are given towards the end of this section. 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 can't do.

Making duplicate copies of disk files

a. **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 a single wildcard file name template and you don't want to store the copies in the same directory as the original files. If you want to make duplicate copies of files in the same directory, you have to copy the files one at a time.

b. **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 don't already have any files with this filename and filetype 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. Use the DIR command (section 2.4.1) both to find out the size of the files you want to copy and to find out if there is enough room on the destination disk for the new files. The free space on a disk is always given as part of the directory information.

c. **Work out how to tell MS-DOS the directory holding the source file(s) and the destination directory.**

If you are going to have a sequence of command lines making copies in your destination directory of files in your source directory, it can save you a lot of effort if both directories are current directories.

d) **Type in the appropriate COPY or XCOPY command.**

If you haven't set up your PPC to automatically verify everything that is written to a disk (see section 2.4.1), we suggest you use these commands' / V option. This tells MS-DOS to verify that the new file is an exact copy of the original file and that it can be read back by checking, for example, that it doesn't have any bad sectors in it.

The new file - the Destination file - is completely independent of the source file once the copy has been made and can be sent to any part of your system.

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 monitor 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 of telling MS-DOS that, for example, the keyboard is the source and the printer is the destination is simply to 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 AMSTRAD PPC's input and output devices are listed in Appendix D.

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 do not have to be stored on 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 towards the end of this section.

2.5.2.2 Examining text files

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 PPC's screen; the PRINT command is used to output the file on a printer attached to your PPC.

The TYPE command can only display one file at a time and unless you make the display pause, for example by using the MORE filter (described in section 2.4.3), 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 thought.

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 at the end of this section, where the PRINT command is described in detail.

2.5.2.3 Deleting files

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 aren't 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 don't want are deleted. The way to ensure this is to write-protect all important files (see section 2.5.3).

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 1.7.5). 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 didn't mean to delete.

2.5.2.4 Renaming files

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 mustn't use a name that has already been given to another file or to a directory within the same directory.

Renaming a file doesn't 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.

2.5.2.5 Comparing files

MS-DOS 3.3 provides two file-comparison commands, **COMP** and **FC**. The **FC** command provides a number of options which are not available in the **COMP** command and can be used to display the differences between files directly, or to store them for subsequent examination. It operates in the same way as the **COMP** command supplied with the AMSTRAD PC1512 and 1640, running MS-DOS 3.2.

COMP is used simply to check that two files are identical. You would use it, for instance, to check that a copy you had just made was identical to the original.

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**
- * **byte-by-byte comparisons to compare versions of a machine-code 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.

2.5.2.6 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 2.4.1.1) 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.

2.5.3 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 PPC. 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.

2.5.3.1 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 'backing up'. There are a number of possible ways of making back-up copies of files:

- you can make copies of whole disks, by using the DISKCOPY command (see section 2.6.2) or the XCOPY command (see section 2.5.2.1)
- you can copy individual files onto a different disk, by using the COPY command or the XCOPY command (see section 2.5.2.1)

2.5.3.2 Controlling the creation of security copies

If you have a group of files you back up regularly, it saves you time if you only back up those files in the group which have 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.

To get a similar effect when your disk was created by a DISKCOPY or several COPY commands, you need to use the REPLACE command.

Options offered by the REPLACE command include:

- * **spotting brand new files and selectively adding these to the set on the back-up disk**
- * **updating read-only files as well as unprotected files**

From time to time when using the XCOPY command, you may want to include in the back-up a particular file even though it hasn't been changed, or to exclude a file even though it has been changed. For this, you need to change the file's Archive attribute with the ATTRIB command. (An attribute is a kind of switch on a file which can be set to 'On' or 'Off'.)

- If you want the file to be backed-up, you should set the switch to On (+A)
- If you don't want the file to be backed-up, you should set the switch to Off (-A)

2.5.3.3 Protecting against accidental deletion

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., except by deliberate use of the /R option with the REPLACE command

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.

2.5.3.4 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 section 2.4.1.1).

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 PPC's internal clock set correctly. The clock setting is held in a part of the AMSTRAD PPC's memory that is powered by the internal batteries, even when the power selection switch is in the BATT.OFF position; so keeping the clock right means either changing the batteries when the battery low alarm is heard, before they go flat, or renewing them when, on startup or reset, you see that the date has been reset to January 1980, or are prompted to set the time and renew the batteries; then using the TIME and DATE commands to reset the calendar (see section 2.7.3).

Note:

The correct time and date are maintained during battery replacement if the PPC is kept connected to one of the approved external power sources, regardless of the setting of the power selection switch.

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 that 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 hasn't been 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

Note:

You can use a file name template, constructed using the wildcard characters * and ? (see 'Simple commands' in section 2.2.2) 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 2.4.2.4)

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 *.*
A          A:\COMMAND.COM
A          A:\4201.CPI
A          A:\5202.CPI
A          A:\ANSI.SYS
A          A:\APPEND.EXE
A          A:\ASSIGN.COM
A          A:\ATTRIB.EXE
A          A:\CHKDSK.COM
A          A:\COMP.COM
A          A:\COUNTRY.SYS
A          A:\DISKCOMP.COM
-          .....
A>
```

- * Display the attributes of all the *.TXT* files in *\DIR1* (a subdirectory of the Root directory) with the command:

```
ATTRIB \DIR1\*.TXT
```

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 root 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 X
```

```
File1 = YY
```

```
File2 = ZZ
```

where 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

Internal command

```
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 monitor 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, though 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 1.7.5). 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][/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
```

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	Monitor screen
PRN or LPT 1	A printer attached to the parallel printer port on the back of your PPC
AUX or COM 1	The Serial Interface
NUL	A 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 COM 1	The Serial Interface

Notes:

- i) In general, everything is recorded in the destination file until Ctrl-Z is entered. The exception are line-editing keys such as [F1] and [←] used when typing information at the keyboard.
- ii) 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...*][*d* :][\][*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*

Notes:

- i) You can use a file name template (see section 1.7.5) 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.
- ii) 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 Ctrl-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:

- i) 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.
- ii) 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, that is, that your new template doesn't lead to filenames more than 8 characters long or filetypes more than 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.

DEL

Internal command

ERASE

```
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
```

Note:

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 B, your command line would become:

```
DEL B: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 won't necessarily clear the directory completely: any subdirectories will be unaffected.

`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 (that is, 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 AMSTRAD PPC text editor, RPED (see section 1.6.8).

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 doesn't 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 2.4.2) 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 [↵] 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 doesn't 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 isn't 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 2.5.2.6).

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]

Note:

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 don't 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 haven't so far 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*).
l specifies the line before which the copy is to be inserted (which can't 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,IM** 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 *l*.
- [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 don't 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][?]Rold-string[Ctrl-Z[new-string]]**
 Replace all occurrences of *old-string* in a range of lines with *new-string*. If *new-string* is omitted, EDLIN will delete *old-string* (ie. it won't 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][d :][\][*path \ filename . filetype*

Transfer the whole of the named file into the workspace, inserting it immediately before Line *l*. If *l* 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

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.

FC *[option[option...]][d:][\][path\]filename . filetype*
[d:][\][path\]filename . filetype

Compare files

The FC command is used to compare two files.

The files for comparison are:

- either treated as text files and compared line-by-line
- or treated as binary files and compared byte-by-byte

When a difference is found between two text files, FC reads on looking for lines from the two 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.

FC can only read on so far. All the lines it is trying to match are stored in an area of your PPC's memory described as the FC command's buffer. If the buffer becomes full before any match has been found, FC declares the files 'too different' and ends the comparison. You can control how big this buffer is.

What type of comparison is made is controlled by which options are selected. 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 file have to be identical for the comparison to resynchronise

The way FC reports the differences is shown through the examples at the end of this description.

* To 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, just display the last line of previous matching section and the first line of the following matching section - not all the lines in between. (The lines in between are represented by ...)

/B Forces a binary comparison of both files. FC compares the two files byte-by-byte, with no attempt to resynchronize after a mismatch. The mismatches are displayed in the format:

Compare error at OFFSET X

File1 = YY

File2 = ZZ

where YY and ZZ are the hex codes for the unequal characters.

/C Treat all letters as upper case in the comparison.

/L Force a line by line comparison. (Used when the filetype of either file is EXE, COM, SYS, OBJ, LIB or BIN.)

/LBN Set the buffer size to n lines. (The default is 100 lines; the maximum depends on the amount of memory space your PPC has.)

/N Add line numbers when reporting where the differences 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 is 2 lines.)

Notes:

- i) If the number of consecutive lines that are different is greater than the buffer space, FC won't be able to resynchronise the comparison. If this happens, the message Resynch failed. Files are too different will be displayed and then the comparison will be abandoned. Use the **/LBN** option to increase the size of the buffer before comparing these files again.
- ii) You can't 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

* To compare files byte-by-byte

Form FC **[/B]**[d:][\][path\]filename . filetype **[d:][\][path\]filename . filetype**
The file being compared *'Standard' for the comparison*

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 line-by-line the two text files, MYFILE.1 and MYFILE.2, you might use the command line:*

```
FC MYFILE.1 MYFILE.2
```

(assuming that the external command FC is stored either in the default directory or in a directory that MS-DOS automatically searches - see section 2.4.2.4)

This would compare 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
Line 4
***** myfile.2
Line 4
Line 5
Line 6
Line 7
Line 8
*****
A>
```

If the differences were rather more extensive, such a report would prove very long. In that case, you should use the /A option which just gives the last line of the previous matching section and the first line of the following matching section, as follows:

```
FC /A MYFILE.1 MYFILE.2

A>fc /a myfile.1 myfile.2
***** myfile.1
Line 4
***** myfile.2
Line 4
...
*****
A>
```

- * *If the differences were very great, you would see the message:*

```
Resynch failed. Files are too different
```


If you wanted to know what the differences were, you would have to re-do the comparison, this time increasing the number of lines that FC can compare at a time. The default number of lines is 100, so you might use the command line:

```
FC /A/LB200 MYFILE.1 MYFILE.2
```

- * *None of these comparisons tell you whereabouts in the files the difference was found. To discover that, you need the /N option and a command line like:*

```
FC /N MYFILE.1 MYFILE.2
```

- * *To compare the files COMMAND.EXE and COMMAND.SYS, you might 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 this:

```
A>fc command.com command.sys
fc: command.sys longer than command.com
```

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

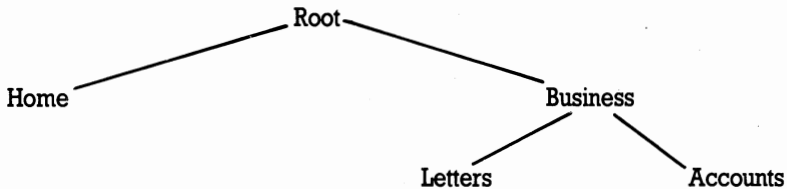
Form MKDIR [d:][\]path
or MD [d:][\]path

Note:

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 don't 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.

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 PPC 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 PPC, there are a number of other parameters which you can set, including:

- the device name your printer has (see section 2.7.1)
- the size of the print buffer
- how many files can be waiting to be printed
- how much of your PPC's processing time is given over to printing the files

Notes:

- i) 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.
- ii) 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.
- iii) 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.
- iv) 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 ...][
File or files to be printed option[option...][
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 don't 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 [←]. 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, that is, your PPC 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 PPC was doing.

/ U : busyticks

Sets how long your PPC 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 PPC 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:

- i) Be careful only to name an output device attached to your PPC as the output device for the PRINT command.
- ii) 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 PPC.
- iii) 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 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 2.4.2.4)

If this is the first time you have used the PRINT command since switching on or resetting your PPC, the following message will appear on your screen:

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

This message won't 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 PPC ([Ctrl][Alt][Del]) before using this command line.

You mustn't 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 didn't 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 didn't 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.

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 RENAME [*d:*][\][*path* \]*old-name new-name*

or REN [*d:*][\][*path* \]*old-name new-name*

where *old-name* is:

filename . filetype

new-name is:

filename . filetype

Notes:

- i) The wildcard characters * and ? may be used to specify a group of files in the same directory (see section 1.7.5). However their use in giving the new file template must be identical to their use in giving the old file template.
- ii) 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 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 \DIR1 directory on Drive B (a subdirectory of the Root directory), the command line would be:

```
REN B:\DIR1\NEWDATA.1 OLDDATA.1
```

Notice that you don't 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 versions 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 didn't 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 2.4.4.3).

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 either the /D or 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 replacing and adding files (that is, allows you to swap the disks in your floppy disk drives after issuing the instruction)

Exit codes

- 1 Command line error, for example, 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, that is, you tried to update a write-protected file.
- 8 Insufficient memory for REPLACE to operate
- 15 You specified a drive that doesn't exist

Notes:

- i) If you don't specify the /R option, any attempt to replace Read-Only files will cause the insert process to stop.
- ii) If you don't specify the /W option, the process of replacing and adding files will start immediately working with the disks currently in the drive(s).
- iii) You cannot use REPLACE to update the hidden system files. Use the SYS command to do this (see section 2.6.1).

REPLACE can return standard MS-DOS error codes as well as the specific error codes listed above.

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 2.4.2.4)

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 - that is, 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 to about each file before it copies the new version to your back-up disk, you need to use the /P option - that is, a command line like:

```
REPLACE *.TXT B:/A/P
```

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.

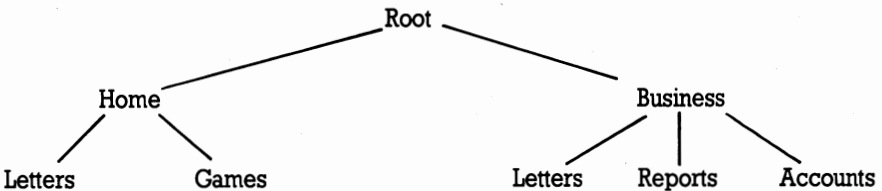
Form RMDIR [d:][\]path
or RD [d:][\]path

Notes:

- i) 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.
- ii) 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 2.4.1.2).
- iii) Include the optional backslash if the path starts at a root directory; don't include it if the path starts at a current directory.

Examples:

Suppose the pattern of directories on the disk in the default drive 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 directories.
RD GAMES
RD LETTERS
CD \               - Change to a different current directory (you can't
                  delete the current directory)
```

```
DEL HOME\
RD HOME           - Clear and remove the Home directory
```

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 don't 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 2.4.2.4)

The result could be something like this:

```
A>tree b:
```

```
DIRECTORY PATH LISTING FOR VOLUME AMSTRAD2
```

```
Path: \SUB1
```

```
Sub-directories: SUB2
```

```
Path: \SUB1\SUB2
```

```
Sub-directories: None
```

```
A>
```

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 B, use the command line:*

TREE B:/F

The result could be something like this:

DIRECTORY PATH LISTING FOR VOLUME AMSTRAD2

Files: ROOTFILE

Path: \SUB1

Sub-directories: SUB2

Files: SUB1FILE

Path: \SUB1\SUB2

Sub-directories: None

Files: SUB2FILE

TYPE

Internal command

TYPE [d:][\][path\]filename . filetype

List a simple text file.

TYPE displays on your screen the contents of the named file. 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:

```
A>type autoexec.bat
PATH \;
KEYB US 437
ECHO OFF
DATETEST
IF ERRORLEVEL 1 GOTO SIGNON
DATE
TIME
:SIGNON
CLS
VER
LCD
A>
```

* If the *AUTOEXEC.BAT* file you want to display is in the Root directory on Drive B, the command line you need is:

TYPE B:\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:

A>type example

The AMSTRA-PP is a compact computer which is easily transportable

between a

n

number of locations for use in a variety of ways

For

example it could help

=

.....

A>

VOL

Internal command

VOL *d*:

Display disk volume label

VOL displays a DOS disk's label, that is, the name by which the disk can readily be identified and which you set by using the LABEL command (see section 2.7.2).

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, and the directories that can be copied are ones that branch off the source directory. Empty directories can be copied as well as ones 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 2.4.4.3).

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

- | | |
|--------------------------|---|
| <code>/A</code> | Only copy files that have the Archive attribute set (see section 2.5.3.3) Note: The Archive attribute isn't reset after copying (see the /M option) |
| <code>/D:mm-dd-yy</code> | Only copy files that have been changed on or after the given date |
| <code>/E</code> | Copy empty subdirectories as well as ones with files in them. Note: The /S option must be selected as well. |
| <code>/M</code> | Only copy files that have the Archive attribute set (see section 2.5.3.3) Note: The Archive attribute is reset after copying (see the /A option) |
| <code>/P</code> | Confirm each copy |
| <code>/S</code> | 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)

Exit codes:

- 0 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 wasn't found
- 5 Copy terminated by user after a disk error

Notes:

- i) 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 assumed as the destination directory. *.* is the default file name template for both the source and the destination. If you don't 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.
- ii) 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 1.7.6).

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 B, you might use the command line:

```
XCOPY \DIR1\*.* B:\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 2.4.2.4)

* 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 PPC, 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 rearmost disk drive) for the destination disk.

If you wanted to copy empty subdirectories as well as ones with files in them, your command line would become:

```
XCOPY \DIR1\*.* B:\DIR2\ /S/E
```

* *If you have a single-drive PPC 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 command 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).

2.6 PROCESSING FLOPPY DISKS

Processing your floppy disks covers:

* **Preparing new blank disks**

- Dividing up the blank disk into sections so that your PPC 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.

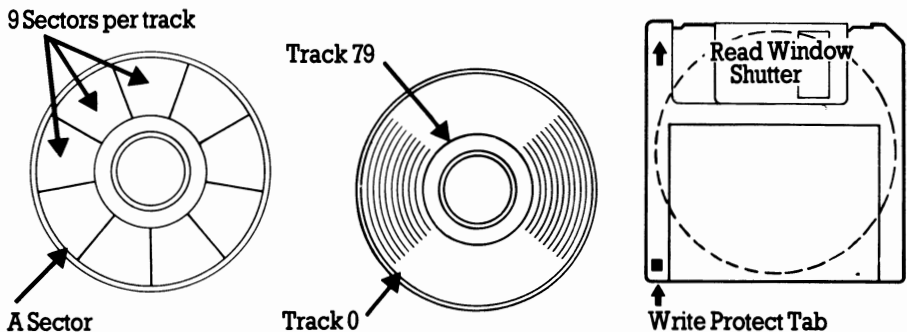
These tasks are fundamental to using floppy disks in your PPC. For example, the storage area of each new blank disk you buy must be formatted before you can use it, even if you just want to store one program or data file on it, and the process of copying a disk automatically includes formatting the disk 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 section 1.6 - the section of this manual that gives step-by-step instructions in using MS-DOS for the most common tasks on your PPC. If such step-by-step instructions are what you are looking for, we suggest you turn to that section. Here, we describe some special features you can build into these processes.

2.6.1 Preparing new disks for use (Formatting)

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 you use to format the disk in. For the standard drives supplied with your AMSTRAD PPC, the appropriate format is 80 tracks (on both sides of the disk) and 9 sectors per track.



The command can also be used with an old disk on which the data has become corrupted. If the disk hasn't been physically damaged, formatting the disk can reclaim all its storage space. However, before you do this, do 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, that is, as a disk that you can insert immediately after you switch your PPC on**
- * **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**

Notes:

- i) When you specify that you want the disk to be used either now or later as a startup disk - that is, as a disk that you can use immediately after switching your PPC on, `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.
- ii) Giving a disk a label gives it a name that will be shown on the screen every time you use the `DIR` command to display details of files on the disk.

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 by about a sixth. 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.

2.6.2 Copying disks

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; however, any processing you do of either disk will stop them from being identical.

The command used is `DISKCOPY`.

The only information that DISKCOPY needs is which drive will hold the disk you want to make a copy of (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 at the point in the process at which you need to change the disk in the drive from the source disk to the target disk.

Notes:

- i) 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.
- ii) In order to copy your disks as efficiently as possible, the DISKCOPY command uses all of the PPC's available memory, including that which is normally used to hold the internal commands. When DISKCOPY is terminated it will therefore 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> prompt. (No message will appear if COMMAND.COM is on the disk already in drive A; it will be reloaded automatically).

2.6.3 Checking disks

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 the areas of the disk 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.)

WARNING

Do not use the CHKDSK command if you are running the MIRROR II communications software in 'background' mode. (See section 3.6.1.9 for an explanation of using MIRROR in this way). MS-DOS can only correctly examine the disk if there are no other programs which might be reading or writing files at the same time as it is checking the disk.

2.6.4 Comparing disks

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 which you copy the disk to alternately and so 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 specifying on which side of the disk and in which track it found the discrepancy. Either one side or both sides of the disks can be compared in this way. Full details are given in the reference part of this section.

Notes:

- i) 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.
- ii) In order to compare your disks as efficiently as possible, the DISKCOMP command uses all of the PPC's available memory, including that which is normally used to hold the internal commands. When DISKCOMP is terminated it will therefore 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> prompt. (No message will appear if COMMAND.COM is on the disk already in drive A; it will be reloaded automatically).

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.

WARNING

Do not use the CHKDSK command if you are running the MIRROR II communications software in 'background' mode. (See section 3.7.1.9 for an explanation of using MIRROR in this way). MS-DOS can only correctly examine the disk if there are no other programs which might be reading or writing files at the same time as it is checking the disk.

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 n nnn.CHK. If these broken files are text files (or word processor files), you can then use a text editor or 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 CHKDSK *[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 the /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 2.4.2.4)

- 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 b:
```

```
730112 bytes total disk space
30720 bytes in 28 user files
699392 bytes available on disk

655360 bytes total memory
593744 bytes free
```

```
A>
```

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
Volume PPC512-640 created DEC 1, 1987 11:33a

730112 bytes total disk space
53248 bytes in 3 hidden files
1024 bytes in 1 directories
349184 bytes in 44 user files
326656 bytes available on disk

655360 bytes total memory
593744 bytes free
```

All specified file(s) are contiguous

```
A>
```

DISKCOMP

External command

DISKCOMP *source-drive* : *target-drive* : [/ 1] [/ 8]

Compare contents of target drive against contents of source drive

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.

Note:

In order to compare your disks as efficiently as possible, the DISKCOMP command uses all of the PPC's available memory, including that which is normally used to hold the internal commands. When DISKCOMP is terminated it will therefore 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> prompt. (No message will appear if COMMAND.COM is on the disk already in drive A; it will be reloaded automatically).

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:

- i) If you don't specify a target drive, this is taken to be the default drive.
- ii) 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 target disk as it compares the disks section by section.
- iii) You cannot use DISKCOMP on a drive that has been ASSIGNED, JOINED or SUBSTITUTED (see Section 2.4.1).

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 2.4.2.4)

This is the command line to use if Drive A 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
```

```
Compare another diskette (Y/N) ?n
```

```
A>
```

* If you just want to compare one side of each disk, use the command line:

```
DISKCOMP A: B:/1
```

DISKCOPY

External command

DISKCOPY *source-drive* : *target-drive* :

Copy contents of source drive to target drive

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.

Note:

DISKCOMP and DISKCOPY apply only to floppy disks. You must use COPY, BACKUP and RESTORE to make copies between hard disks and floppy disks.

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.

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.

Note:

In order to copy your disks as efficiently as possible, the DISKCOPY command uses all of the PPC's available memory, including that which is normally used to hold the internal commands. When DISKCOPY is terminated it will therefore 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> prompt. (No message will appear if COMMAND.COM is on the disk already in drive A; it will be reloaded automatically).

Form DISKCOPY [*source-drive* :] [*target-drive* :]

Notes:

- i) If you only specify one drive, then the target drive is taken to be the default drive. If you don't specify any drives, DISKCOPY performs a single-drive copy that uses the default drive as both the source drive and the target drive.
- ii) 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.
- iii) The target disk will be formatted before the copy is made if it hasn't been formatted already.

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 2.4.2.4)

This is the command line to use if Drive A 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 PPC, 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, that is, the disk you want to store the copy on). If you have a two-drive system, put the target disk in Drive B (the rearmost disk drive).

If the target disk has not been formatted, you will see the following message on the screen:

```
Formatting while copying
```

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**
- * **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 AMSTRAD PPC will be formatted to have 80 tracks (on both sides of the disk) and to have 9 sectors per track.

The disk can be formatted:

- to be used to store files and programs
- to be used as a Startup disk (that is, it can be used immediately after the PPC is switched on)

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 2.4.4.3).

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 cover the write-protect notch on the disk holding the FORMAT program: even the most careful user can make typing mistakes!

In order to create new MS-DOS Startup disks, you should insert your SYSTEM DISK 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

- /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 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.

Exit codes:

0	Format successful
3	Terminated by user
4	Fatal error

Notes:

- i) If you have the **FORMAT** external command stored on a floppy disk, write-protect this disk by sliding the write-protect shutter toward the back of the disk (away from the arrow) to leave the window **OPEN**, before you type in a **FORMAT** command.
- ii) 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
- iii) 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 **AMSTRAD PPC SYSTEM** disk
- iv) You should not use the **FORMAT** command with any **ASSIGNed**, **JOINed** or **SUBSTITuted** drives (see section 2.4.1)

Examples:

- * *To format a floppy disk for 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 isn't already) and use the command line:*

```
FORMAT B:
```

If you have a single-drive PPC, 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, that is, to use to load the **MS-DOS** operating system from immediately after switching on or resetting your PPC, put a copy of the **AMSTRAD PPC** system disk in Drive A, make Drive A the default drive (if it isn't already) and use the command line:*

```
FORMAT B:/S
```

If you have a single-drive PPC, 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.*

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:

- i) If you want to update your system files but the new IO.SYS and MSDOS.SYS files take up more room than the old ones (for example, Version 3.3 files take up more room than Version 2.0 files that have been prepared on another compatible computer) you will have to re-format the disk. **Remember to copy all the data and program files from this disk to other disks before re-formatting the disk.**
- ii) 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 2.4.2.4)

2.7. TAILORING YOUR PPC TO YOUR NEEDS

When you bought your AMSTRAD PPC, it was set up to be used in a very conventional way - with the keyboard your main means of giving instructions and information (that is, input) to the computer and the monitor screen used by the computer to display (that is, 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 section describes how to set up your PPC to work in precisely the way you want.

You might, for example, want to set up your PPC so that it can send the data it normally sends to the printer down a communications link to another computer. Or 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 PPC to your needs by:

*** Setting up Input and Output Devices**

Select a 40 or 80 column monitor screen width. (The 40 column option is not available in MDA mode. Please refer to the instructions for connecting an external monitor, in appendix E).

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 PPC**

- by setting up a personalised system prompt
- by giving disks labels

*** Setting your PPC's clock**

- setting the date
- setting the time

2.7.1 Setting up your PPC's Input and Output Devices

When you buy an AMSTRAD PPC, it is 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 AMSTRAD PPC**

This is a perfectly satisfactory arrangement, but there are two possible ways in which you might want to change it:

- a. 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 PPC 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.
- b. 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 PPC.

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 2.4.3 covered 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

Normally, all command lines are typed in at your PPC's keyboard. However, if you link your AMSTRAD PPC to another computer or if you attach a terminal to your computer, you might well want to give your PPC instructions from the other computer or from the terminal.

You probably will have connected the computer or the terminal to your PPC via the Serial Interface on the back of your AMSTRAD PPC. 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 PPC by holding down [Ctrl] and [Alt] and pressing [Del] (using your PPC's keyboard) or switching off and switching on again. Either of these actions will, however, lose any unstored data in your PPC.

Sending printer output to the Serial Interface

Printer output from your programs is normally sent to the Parallel Printer port on the back of your AMSTRAD PPC. 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 output 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 AMSTRAD PPC. 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

Your PPC must be set up to communicate with each of its input and output devices. MS-DOS initially holds the details your PPC needs to communicate with the devices it is supplied with but these details have to be changed when:

- * **you install a new peripheral**

For example, you might install an external monitor, 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. (Though MIRROR II users can set all necessary communications parameters via the MIRROR software).

- * **you run a program that has been set up for a different screen**

To set up a parallel printer

The aspects of a parallel printer you may have to set are the maximum number of characters that can be printed on a line and the spacing between the lines. In addition, because your PPC 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

PPC 640 users will find all necessary information on the use of the integral modem and the MIRROR II software in section 3. The integral modem uses its own internal interface, COM 2.

An external modem may be connected to the serial port (COM 1) of any model of the AMSTRAD PPC, in which case you may need to use the MODE command to set the following parameters.

- * **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**

The values you need to set should be specified in the literature supplied with the communications link. (The information in section 3 is intended to cover the use of MIRROR II, as supplied by AMSTRAD, and the PPC 640 integral modem. If you wish to use another modem or software package, you should obtain the relevant information from the dealer who supplies them.)

MS-DOS has built into it some default settings for the Serial Interface on the back of your PPC which may well suit an externally connected 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.

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 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

The AMSTRAD PPC normally has an 80-column display and can drive either a colour or a monochrome screen. (The LCD will work correctly whether the PPC is set up for colour or monochrome output, but the 40 column screen width option is NOT available in monochrome mode. Please refer to the instructions for connecting an external monitor to the PPC, in appendix E).

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 the options which you are currently using, they may not give quite the screen output you expected. To correct this, you will need to set your PPC'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 PCs, changing screen mode in this way won't affect the screen alignment and so no adjustment will need to be made.

2.7.2 Personalising your PPC

Personalising your PPC is about changing some aspects of the way your computer works so that, instead of being as provided by AMSTRAD, they are set up to suit you.

Personalising the system 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 PPC is reset. Then MS-DOS will revert to the standard A> 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 2.4.4).

Giving disks labels

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 to give a disk such a label (or to change the label if it already has one).

2.7.3 Setting your PPC's clock

The AMSTRAD PPC's internal clock is maintained in the battery-backed area of the computer's RAM so that it keeps perfect time even when your PPC is switched off. However, it will be reset to 01-1-1980 if the batteries start to go flat. You can avoid this ever happening by changing the batteries as soon as possible once the battery low warning alarm sounds- but if the batteries do go flat, you should first replace them with new ones (see section 1.2.2) and then reset the clock.

You can reset the clock using the MS-DOS commands DATE and TIME, described in this section.

CTTY

Internal command

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 in future expect all command lines to come from a different input device - for example, the PPC'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:

- i) The changeover to using the other device happens immediately.
- ii) The command 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 PPC by holding down [Ctrl] and [Alt] and pressing [Del] on your PPC's keyboard.
- iii) Some programs take their input and send output directly from and to your PPC's hardware. The CTTY command has no effect when this is the case.

DATE

Internal command

DATE [mm-dd-yy]

Set or read the date

DATE is used either to display or set the current date. This date is maintained in battery-backed RAM in your PPC and it is automatically updated every time the PPC's clock gets to 00:00 (that is, midnight). However, if the battery power falls, the clock will be reset to 01-1-1980. You can avoid this ever happening by changing the batteries as soon as possible once the battery low warning alarm sounds- but if the batteries do go flat, you should first replace them with new ones (see section 1.2.2) and then set the date.

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 mm-dd-yy

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

Notes:

- i) 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 [↵]
- ii) For dates after 1999 all 4 digits of the year must be entered.

Example:

To set the date to be November 1, 1989, either use the command line:

```
DATE 11-01-89
```

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 Fri 10-28-1989  
Enter new date (mm-dd-yy):
```

Type:

```
11-01-89 [↵]
```

to set the date.

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 way down 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 don't 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:

- i) You can use any characters in the disk label except:
* ? / \ | . , ; : + = < > []
- ii) Spaces are allowed. If you try to use any of these characters, your new label won't be accepted and you will see an error message.
- iii) You can't label a disk if the drive it is in has been ASSIGNED to another drive or JOINED to a directory.
- iv) 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:

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

To set the new label, type:

```
RECORDS86-7 [↵]
```

To check that the disk has been labelled, type:

DIR B:

at the A> prompt and the message:

Volume in drive B is RECORDS86-87

will be displayed, followed by the root directory listing.

(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 2.4.2.4.)

MODE *device* : *setting* [, *setting*...]

Set mode of operation of Input and Output Devices

The MODE command is used:

- * to set the number of characters per line and lines per inch on a line printer
- * to set the transmission rate, parity etc. of communications links
- * to adjust the screen display
- * to redirect printer output to the serial interface

Notes:

- i) The device name to use for the Parallel Printer on the AMSTRAD PPC is LPT1. Do not use the alternative name PRN in a MODE command.
- ii) The device name to use for the Serial Interface on the AMSTRAD PPC is COM1. Do not use the alternative name AUX in a MODE command. The device name COM2 is used for the integral modem interface, which is not available for use with other devices.
- iii) Error messages will be displayed if you try to set device parameters to invalid settings or you try to direct printer output to a communications device that doesn't exist.

* To set up a parallel printer

Form **MODE** *device* : [*characters-per-line*] [, [*lines-per-inch*] [, *P*]]

where *device* is: LPT1, LPT2 or LPT3

characters-per-line is: either 80 or 132 (the default is 80)

lines-per-inch is: either 6 or 8 (the default is 6)

Notes:

- i) The default value for a particular setting is used if this is either omitted or the setting you give is invalid.
- ii) If you want to use all but one of the default settings, put a comma in for each of the parameters you want the default setting for up to the parameter you actually want to set. For example, if you just want to set the number of lines per inch (the second parameter), use a command line of the form:

```
MODE LPTn : , lines-per-inch
```

Use the P setting if your PPC can send data to your printer very much faster than the printer can output it on paper. The P setting instructs MS-DOS to try to send output to the printer continuously. However, if there is no response from the printer, for example because the printer is off-line, the message

Infinite retry on parallel printer timeout
will be displayed (see appendix G: Troubleshooting).

If you set either the number of characters per line or the number of lines per inch, messages confirming that these values are now in force are displayed as they are set.

The message `LPTn: not redirected` is displayed when you use this version of the `MODE` command, to tell you that any previous redirection to a serial port has been broken.

*** To set up a communications link, via a modem attached to the serial interface**

Form `MODE device: baud[, [parity][, [databits][, [stopbits][, P]]]]`

where *device* is: COM1 or COM2

baud is: 110, 150, 300, 600, 1200, 2400, 4800 or 9600

parity is: N (that is, none), O (that is, odd) or E (that is, even)

databits is: either 7 or 8

stopbits is: either 1 or 2

Notes:

- i) *baud* sets the transmission rate and must be included. At least the first two digits of the *baud* setting must be entered.
- ii) The default value for a particular setting is used if this is either omitted or the setting you give is invalid. For COM1, these are Even parity and 7 databits. If *baud* is 110, then the default number of *stopbits* is 2. Otherwise the default number of *stopbits* is 1.
- iii) If you want to use all but one of the default settings, put a comma in for each of the parameters you want the default setting for up to the parameter you actually want to set. For example, if you just want to set the number of *stopbits* (the fourth parameter), use a command line of the form:

`MODE COMn: baud , , , stopbits`

Use the *P* setting if you want to use the communications port for a serial printer (see 'To redirect printer output to a serial interface' below).

Note: MS-DOS will continuously try to send output to the printer. However, if there is no response from the printer, for example because the printer is off-line, the message

`Infinite retry on parallel printer timeout`
will be displayed (see Appendix G: Troubleshooting).

*** To adjust the screen display**

Form `MODE display`

Options

`4 Ø` 40 characters per line

`8 Ø` 80 characters per line

Notes:

- i) The facilities of the MODE command to adjust the alignment of the screen are disabled on the AMSTRAD PPC. No such adjustment should ever be necessary.
- ii) The 40 character width option is not available with the AMSTRAD PPC if MDA mode has been selected (to drive an external monochrome monitor).

*** To redirect printer output to a serial interface**

To make the redirection:

Form MODE LPTn:=COMm:
where LPTn is: LPT1, LPT2 or LPT3
COMm is: COM1 or COM2

To go back to normal operation:

Form MODE LPTn:[setting[, setting...]]

Notes:

- i) This redirection is used to send printer output to a serial printer, attached to a serial interface. To redirect output normally sent to the Parallel Printer port on the back your AMSTRAD PPC to the Serial Interface (also on the back of your machine), use the command line:

```
MODE LPT1:=COM1:
```

The message LPTn: redirected to COMm: is displayed to confirm the redirection.

- ii) If your parallel printer needs any special settings, set these when you return printer output to the parallel printer port, even if you have set them before. Otherwise the default settings will be used.

Examples:

- * *To set the line pitch (the number of lines per inch) on the parallel printer attached to the Parallel Printer port to 8, use the command line:*

```
MODE LPT1: ,8
```

(assuming that the external command MODE is stored in the default directory or in a directory that MS-DOS automatically searches - see Section 2.4.2.4)

MODE will respond with the message:

```
Printer lines per inch set
```

(PPC640 users will probably prefer to use the supplied MIRROR II communications package with their integral modem, to set up and run their communications tasks, although there is no reason why they should not connect an external modem to the serial port and use these or any other software commands to control it, if they wish. PPC 512 users who wish to use communications will need to connect an external modem to the serial interface connector and provide the software necessary to drive it.)

* *To set the screen display to 40 characters per line, use the command line:*

```
MODE 40
```

* *To redirect printer output to a serial printer attached to the Serial Interface, use the command line:*

```
MODE LPT1:=COM1:
```

MODE will respond with the message:

```
LPT1: redirected to COM1:
```

* *To return to sending printer output to the parallel printer port, use the command line:*

```
MODE LPT1:
```

unless your printer needs a special requirement, for example you need to instruct MS-DOS to try sending data to the printer continuously. Then you will need a command line like:

```
MODE LPT1:,,P
```

MODE will respond with the message:

```
LPT1: not redirected
```

to show that the connection to COM1 is no longer in force.

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 (for example, 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, 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:

<code>\$d</code>	<i>To give you the date</i>
<i>A couple of spaces</i>	<i>To separate the date from the time</i>
<code>\$t</code>	<i>To give you the time</i>
<code>\$h\$h\$h\$h\$h\$h\$h\$h</code>	<i>To backspace over the seconds and hundredths of a second in the time</i>
<code>\$_</code>	<i>To give you a new line</i>
<code>\$n</code>	<i>To give the default drive</i>
<code>:</code>	<i>To give a colon character</i>
<code>\$p</code>	<i>To give the current directory</i>
<code>\$g</code>	<i>To give a > character.</i>

The command line to use is therefore:

```
PROMPT $d $t$h$h$h$h$h$h$h$h$_$n:$p$g
```

TIME

Internal command

TIME [*hh:mm:ss.dd*]

Set or read the time

TIME is used either to display or set your PPC's clock. This is a 24-hour clock which is maintained in battery-backed RAM. Its settings are used to time stamp files and it is therefore advisable to check that the clock has not been reset because the batteries are going flat. If this happens, renew the batteries (see section 1.2.2) and then reset the clock.

Any time 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:ss.dd*

where *hh* represents the hour (00...04) *mm* the minute (00...59) *ss* the second and *dd* the decimal parts of the second.

Notes:

- i) If you only enter some of the parameters, the command will be accepted and the remaining parameters set to zero. For example: TIME 11 will set the time to exactly 11 o'clock.
- ii) 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 [↵]
- iii) The clock is set when you press Carriage Return. 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:

```
Current time is 15:31:05.56
```

Type:

```
16:23
```

and press [↵] when it is exactly 4.23pm.

2.8. SUMMARY OF MS-DOS COMMANDS

APPEND	External command (section 2.4.2.4) Set a search path for data files <code>APPEND [d:]\path[;[d:]\path...]</code>
ASSIGN	External command (section 2.4.1.2) Assigns drive letter to another drive <code>ASSIGN requested-drive=searched-drive</code>
ATTRIB	External command (section 2.5.3.3) Set file attributes <code>ATTRIB [+R -R][+A -A][d:][\][path\]filename . filetype</code>
BREAK	Internal command (section 2.4.2.2) Sets how often MS-DOS checks for Ctrl-Break <code>BREAK ON OFF</code>
CHDIR	Internal command (section 2.4.1.2) Change the current directory <code>CHDIR [d:][\]path</code> <code>CD [d:][\]path</code>
CHKDSK	External command (section 2.6.3) Check disk for errors and file for non-contiguous storage areas <code>CHKDSK d:[\][path\][filename . filetype][/ F][/ V]</code>
CLS	Internal command (section 2.4.1.4) Clear the screen <code>CLS</code>
COMMAND	External command (not covered in this manual) Start the command processor from a higher level <code>COMMAND [d:][\][path][device][option[option...]]</code>
COMP	External command (section 2.5.2.5) Compare files <code>COMP [d:][\][path\]filename . filetype [d:][\] [path\]filename . filetype</code>

COPY	Internal command (section 2.5.2.1) Copy files C O P Y [d :][\][path \]source[/ A][/ B] [d :][\][path \] destination[/ A][/ B][/ V]
CTTY	Internal command (section 2.7.1) Change the device used to issue commands C T T Y <i>device</i>
DATE	Internal command (section 2.7.3) Set or read the date D A T E [mm-dd-yy]
DEBUG	Debugging utility (not covered in this manual) D E B U G [d :][\][path \][filename . filetype][argument[argument...]]
DEL	Internal command (section 2.5.2.3) Delete a file D E L [d :][\][path \]filename . filetype
DIR	Internal command (section 2.4.1.1) Display directory D I R [d :][\][path \][filename . filetype][/ P][/ W]
DISKCOMP	External command (section 2.6.4) Compare contents of target drive against contents of source drive D I S K C O M P <i>source-drive</i> : <i>target-drive</i> : [/ 1][/ 8]
DISKCOPY	External command (section 2.6.2) Copy contents of source drive to target drive D I S K C O P Y <i>source-drive</i> : <i>target-drive</i> :
ECHO	Batch subcommand (section 2.4.4.2) Control echoing of commands on the screen E C H O [ON OFF <i>remark</i>]
EDLIN	Text editor (section 2.5.2) Edit text E D L I N [d :][\][path \]filename . filetype[/ B]

ERASE Internal command (section 2.5.2.3)
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 and return to the previous level (if any)
EXIT

FC External command (section 2.5.2.5)
Compare files
FC [option[option...]][d:][\][path\]filename . filetype [d:][\][path\]filename . filetype

FIND External command and external filter (section 2.4.3.3)
Look for a string in a file
FIND [/V][/C][/N]"string" [d:][\][path\]filename . filetype

FOR Batch subcommand (section 2.4.4.2)
FOR is used to execute a number of similar commands.
FOR %%parameter IN (value[value...]) DO command

FORMAT External command (section 2.6.1)
Format the disk in the specified drive
FORMAT d:[option[option...]]

GOTO Batch subcommand (section 2.4.4.2)
Go to a labelled point in the file
GOTO label

GRAFTABL External command (not covered in this manual)
Load additional character table (for use in graphics mode)
GRAFTABL

GRAPHICS	External command (section 2.4.2.3) Prepare your PPC to print graphics screen displays GRAPHICS <i>printer-type[/R][/B]</i>
IF	Batch subcommand (section 2.4.4.2) IF [NOT] <i>condition command</i>
JOIN	External command (section 2.4.1.2) Join a disk drive into another disk's directory structure JOIN <i>d: [d:]\path</i>
KEYB	External command (appendix E) Configure keyboard KEYB US
LABEL	External command (section 2.7.2) Create or change a disk label LABEL <i>[d:] label</i>
MKDIR	Internal command (section 2.5.1.2) Make a new directory MKDIR <i>[d:]\path</i> MD <i>[d:]\path</i>
MODE	External command (section 2.7.1) Sets mode of operation of input and output devices MODE <i>device: setting[, setting...]</i>
MORE	External filter (section 2.4.3.3) Make output be displayed one screenful at a time MORE
PATH	Internal command (section 2.4.2.4) Define a search path PATH <i>[d:]\path[;[d:]\path...]</i>
PAUSE	Batch subcommand (section 2.4.4.2) Pause while processing PAUSE <i>[remark]</i>

PRINT	External command (section 2.5.2.2) Print text files in the background PRINT [d:][\][path \]filename . filetype [[d:][\][path \] filename . filetype...][/ P][/ C][/ T]
PROMPT	Internal command (section 2.7.2) Set up a new system prompt PROMPT <i>prompt-text</i>
REM	Batch subcommand (section 2.4.4.2) Display remark REM <i>remark</i>
RENAME	Internal command (section 2.5.2.4) Rename a file REN [d:][\][path \]old-name new-name
REPLACE	External command (section 2.5.3.2) Insert new files and update old versions REPLACE [d:][\][path \]filename . filetype [d:][\][path \] option[option...]]
RMDIR	Internal command (section 2.5.1.3) Remove a directory RMDIR [d:][\][path RD [d:][\][path
SET	Internal command (section 2.4.1.4) Set a parameter to a string SET <i>parameter=string</i>
SHIFT	Batch subcommand (section 2.4.4.3) Shift the dummy parameters SHIFT
SORT	External command and filter (section 2.4.3.3) Sort data SORT [/R][/+n] <[d:][\][path \]filename . filetype [>destination]

SUBST	External command (section 2.4.1.2) Substitute the name of an imaginary drive for a path SUBST <i>d: [d:][\]path</i>
SYS	External command (section 2.6.1) Copy system files to the disk in the specified drive SYS <i>d:</i>
TIME	Internal command (section 2.7.3) Set or read the time TIME <i>[hh:mm:ss.dd]</i>
TREE	External command (section 2.5.1.1) Display the pattern of directories on a drive TREE <i>[d:][/ F]</i>
TYPE	Internal command (section 2.5.2.2) List a simple text file TYPE <i>[d:][\][path\]filename . filetype</i>
VER	Internal command (section 2.4) Display MS-DOS version number VER
VERIFY	Internal command (section 2.4.1.4) Turn on and turn off verification of all writes to disk VERIFY ON OFF
VOL	Internal command (section 2.5.3.4) Display disk volume label VOL <i>d:</i>
XCOPY	External command (section 2.5.2.1) Copy files and directories XCOPY <i>[d:][\][path\]source [d:][\][path\]destination [option[option...]]</i>

Batch commands

BATCH	Internal commands (section 2.4.4) Run a batch of commands <i>[d :][\][path \]filename [value[value...]]</i>
ECHO	Batch subcommand Control echoing of commands on the screen <i>ECHO [ON OFF remark]</i>
FOR	Batch subcommand FOR is used to execute a number of similar commands. <i>FOR %%parameter IN (value[value...]) DO command</i>
GOTO	Batch subcommand Go to a labelled point in the file <i>GOTO label</i>
IF	Batch subcommand <i>IF [NOT] condition command</i>
PAUSE	Batch subcommand Pause while processing <i>PAUSE [remark]</i>
REM	Batch subcommand Display remark <i>REM remark</i>
SHIFT	Batch subcommand Shift the dummy parameters <i>SHIFT</i>

3. COMMUNICATING WITH MIRROR II

CONTENTS

3.1 Getting started - A quick introduction

3.2 Notes for the Newcomer

3.3 General details of PC2000 modem

3.4 Introducing MIRROR II

3.5 NEWUSER procedure

3.6 General Calling Procedure

3.7 Working with MIRROR II

3.8 Alphabetical list of MIRROR commands

The quick introduction, in section 3.1, is written so that complete newcomers to communications can dial up a number of services and gain practical experience straight away.

Section 3.2 is also intended for new communications users. It aims to outline, briefly, the concepts involved and terms used. Users with communications experience will probably want to skip those sections go to section 3.3.

For those with no previous experience we hope that sections 3.1 and 3.2 will give an insight into the workings of a communications session, sufficient for them to understand what is taking place during the session and to have some idea of what to do when the unexpected occurs.

Section 3.3 gives a brief overview of the PC2000 integral modem, which forms part of the PPC 640 models; full details of the modem and its command set are given in appendix H.

Sections 3.4 and 3.5 give the information needed to use the facilities which AMSTRAD have built into MIRROR II.

For the user who wishes to develop the system further, the basic steps in making a call via the modem are shown in section 3.6, followed by a concise description of MIRROR's command set and method of operation, in section 3.7.

3.1 Getting Started - a quick introduction.

There are a number of files included with the PPC640 which assist you in getting started. These files are shown on the MIRROR status screen.

To connect to any one of a number of Electronic Mail and Information services follow the procedure in section 3.5

A further powerful use of the PPC640 is the transfer of files from one PPC to another.

3.1.1 Demonstration of PPC performing file transfers with another computer.

For the sake of this demonstration it assumed that two PPC computers are used. This will show the principles involved. In practice, one or other of the computers could be substituted by a different machine, and even different communications software. However, this will normally require the involvement of a 'tame expert'.

The object of the exercise is to demonstrate how one PPC can be left, for example at home or in your office, waiting to answer a call from a second PPC located anywhere near a convenient telephone socket. When the connection has been established, files can be sent in either direction. It is easiest to do this demonstration the first time somewhere that there are two telephone sockets next to one-another. It is very instructive to be able to see both computers operating.

By far the most difficult part of demonstrating file transfers is being able to refer to each machine involved without using jargon or becoming confused as to which machine is meant ! In this case write two slips of paper with the words 'HOME' and 'AWAY' and affix them as instructed to the two PPCs. Both PPCs, if using the correct disks, should have a directory called MAILBOX. In this example we will transfer to and from the MAILBOX directory of the HOME PPC and the root directory of the AWAY PPC.

Firstly set up the answering PPC:

Attach the label HOME. The answering PC must be connected to a direct line (that is, not through a switchboard). Having loaded the MIRROR II software type the command.

```
L0 [↵]
```

Identify the number of the 'ANSMODEM' demonstration and type that number, followed by Carriage Return

The PPC will display the following status report on the Command line:

```
Waiting for call... press Esc to return to command mode.
```

(There is no need to type anything further on this PPC - but we will much later describe how you could use the 'background' feature of MIRROR II to continue to use this PPC while waiting for the telephone call to arrive.)

Secondly, set up the calling PPC:

Attach the label AWAY. Having loaded the MIRROR II software type the command:

```
L0 [↵]
```

Identify the number of the 'CALLHOME' demonstration and type that number, followed by [↵]

In order to correctly dial the number, certain questions will be asked:

```
What is the area code and number you wish to contact?
```

```
Are you connected to a switchboard extension (Y/N) ?
```

..and if so:

```
What number (for example, 9) do you dial to obtain an outside line ?
```

MIRROR then assembles a complete telephone number and asks for confirmation:

```
Does this look OK (Y/N) ?
```

If you have decided that the resulting number appears plausible, then the Command Line will display the message **DIALING**, the agreed telephone number and a timer counting down in seconds. On the PPC speaker you should hear:

- the dialling tone
- dialling pulses
- perhaps a very short amount of ringing
- a whistling tone
- perhaps a burst of loud noise
- and finally silence, as the Status Screen disappears and is replaced by an initially clear screen.

What may go wrong.

- the number may be busy (engaged).
- the number may not answer (continually ring).
- the number may not be correctly connected and result in either a number unobtainable tone, or a voice answering the call.

In all these cases, when the on-screen timer has counted down to zero, the call will be terminated and **MIRROR** will then display the normal Command prompt. To re-dial manually, simply press the [←] key once.

If you wish to terminate the call sooner than at the end of the automatic timer count-down simply press [**Esc**].

MIRROR will then display the normal Command prompt. To re-dial manually, simply press the [←] key once.

Using the File Transfer

On the AWAY PPC	On the HOME PPC
<p>Type the password. This is initially set to RORRIM (which is MIRROR backwards!). You will not see the characters as you type.</p> <p>When the message 'Password Accepted' is received from the HOME PPC, the command 'cd \ mailbox' is sent back to the HOME computer.</p> <p>The command 'dir /s' is sent to the HOME PPC.</p> <p>You are asked if you wish to (S)end, (R)eceive or (C)hat. Select 'S'. You are then asked:</p> <p>What file do you want to send (from AWAY to HOME) ?</p> <p>Reply SEND.TST . This is a file in the root directory of the AWAY disk.</p>	<p>The password will appear, character by character, as it is typed. If the password is entered correctly, the message 'Password accepted' is sent back to the AWAY PPC. After three unsuccessful attempts the message 'Illegal Access' is sent and the telephone call is terminated.</p> <p>The HOME computer changes directory to \ mailbox and returns the confirmation message 'Current directory is MAILBOX'</p> <p>Display of the directory, including file sizes is shown and transmitted back to the AWAY PPC.</p>
<p>The file transfer is completed with a progress display on each PPC. Only the sending (AWAY) PC will update the '% complete' box.</p>	<p>This system has been arranged so that no existing files on the HOME PPC can be deleted or replaced by the file transfer process. Therefore if the receiving (HOME) PPC already contains a file called SEND.TST in the MAILBOX directory (perhaps from a previous session) an error will be displayed on the screen. The sending (AWAY) PPC will indicate such a failed transfer by not updating the '% Complete' box - which would read 100 at the end of a successful file transfer.</p>

You are asked if you wish to transfer another file. Reply 'Y'.

You are again asked if you wish to (S)end, (R)eceive or (C)hat. Select 'R'.
You are then asked:

What file do you want to receive
(from HOME to AWAY) ?

Reply FROMHOME.TST . This is a file in the MAILBOX directory of the HOME disk.

The file transfer is completed with a progress display on each PPC. Only the sending (HOME) PC will update the '% complete' box, but the receiving (AWAY) PC will increment the 'Block' counter as the file is received.

You are again asked if you wish to (S)end, (R)eceive or (C)hat. Select 'C'.

The two PPCs are now connected so that everything typed at one is echoed on the other. Two commands are given (each started by pressing [Esc]) which will end the Chat session. Pressing [Ctrl]C on the AWAY computer will allow you to enter a MIRROR command which will then be executed on the HOME computer. Not all commands are allowed (for security reasons).

Changing the password

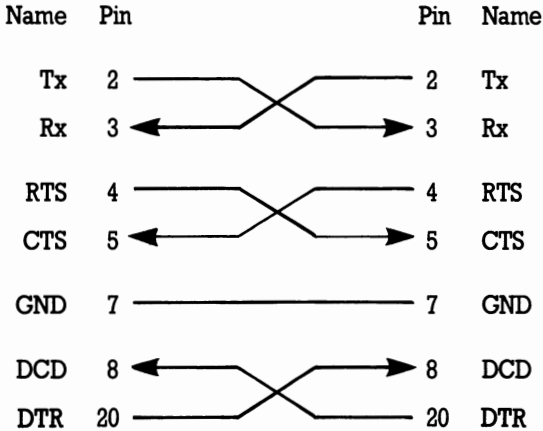
De-activate the HOME PPC by pressing [Esc]. Then type the following at the Command prompt:

```
pw <newpassword>  
sa ansmodem  
e
```

3.1.2 Exchange data with another computer via a direct cable connection.

This can be performed using the same commands as when using two modems. As before, one computer answers and the other, calling computer, issues all the commands.

The recommended cable connection is:



The answering (HOME) computer must run the MIRROR program ANS232 and the calling (AWAY) computer the program FILETRAN. Otherwise, operation is identical to section 3.1.5 above (ignoring, of course, all references to telephones and calls). This method may be used to transfer files between the PPC and computers with 5 1/4 " or hard disks. A kit comprising a stripped-down MIRROR II and the program file 'ANS232' plus a suitable cable is available from Authorized Amstrad distributors.

3.2 Notes for the Newcomer.

This section is intended to introduce newcomers to communications to the basic concepts and terms used in computer communications over the public telephone network. Communications is a complex subject, partly because of its technical nature but also because it has developed in a piecemeal manner, with different techniques being used in different countries and often a variety of methods in use in one country.

It is for this reason that we have supplied you with the MIRROR II software, which will remove much of the difficulty in communicating. The files which we have provided within MIRROR II contain all the commands and settings which can be preset to call the service for which they are written. They will also help you to enter any further inputs which are needed, such as telephone numbers and your name. When, as we hope, you wish to tackle more ambitious projects, these notes may help you to understand what is involved.

We cannot attempt to fully explain the subject of communications in this manual; it would take many volumes of this size to cover that subject alone. However, the following explanation of basic principles and technical terms should enable you to use MIRROR II with confidence.

Some of the commonly used communications terms have more than one meaning; in other cases there is more than one term for the same thing. Because these terms are used in other technical literature which you may read, we have pointed out in the following text those which we think may be confusing or ambiguous, and given alternatives which you may see used elsewhere.

Here are the common terms and concepts which you need to understand:

a. Host

This is one of the terms which has two possible meanings when referring to communications. It can mean the computer to which a particular modem is attached. Alternatively, it can mean the system which is to transmit the data when a transfer takes place. Because it is commonly used in both senses, we have used it in this manual in both ways.

b. Terminal

When a host system sends data, the receiving modem must be in Terminal (listen) mode. The data received will simply pass through the modem to the PPC, treating the PPC as a terminal connected directly to the host system.

c. Emulation

Many terminals are quite specialized and interpret the characters which they receive from the host system in a non-standard way. So that your PPC knows what should really appear on the screen when it receives a stream of characters intended for such a terminal, MIRROR II provides a number of EMULATION programs which take data intended for selected, well-known terminal types and produce the character codes needed to create the correct display on your monitor. So if, for example, you intend to connect to a system which produces output suitable for display on an IBM 3101 terminal, you would use MIRROR II's EMulate command to select the IBM 3101 emulation file before commencing the data transfer.

d. Modems

Telephone systems, being originally developed for speech transmission, use an analogue (or continuously variable) form of signal. Almost all computers use digital (or two state) signals. The job of the modem is to convert the digital signals to analogue signals for transmission down the telephone line. At the receiving end there must be another modem to convert them back again.

The process of converting the signals from digital to analogue and back is known as MODULATION and DEMODULATION. It is from the first letters of each of these words that the term MODEM is formed.

Early models of modem could do little more than that basic task. Now, however, many models have memory and computing features built in, allowing a whole sequence of actions to be carried out in response to a simple command. The modem in your AMSTRAD PPC has been designed to take advantage of the latest developments in circuitry and components. It provides a high level of intelligence, using commands compatible with the widely used HAYES command set.

e. Transmission Modes

It is possible to transmit data along the telephone lines at different speeds. Basically, the higher the speed the faster the transfer, saving both time and cost. But at higher speeds errors are more likely to occur. There is also a maximum speed limitation due to the quality of the line. There are different modes of modulation too. The modes which the modem in your PPC can operate in are listed in appendix H.

The MODE defines the speed of working and the type of modulation. Because the quality of telephone lines varies greatly, it is not always practical to use the higher speeds. On a poor line the error rate may be so high that the advantage of a higher speed is lost, due to the need to re-transmit the faulty information. Some modes therefore offer the option of reverting to a lower speed if the line is poor.

There are an immense number and variety of modems in use throughout the world, each having features and restrictions dependent on the state of development when they were designed and the limitations of cost. Hence the need for a large number of different modes of working in use at the same time. Your integral modem, which adjusts automatically to the speed selected by your PPC, can operate over a very wide range of modes.

All the modes which the PPC modem can operate in are listed in appendix H, section H.5. You will see the terms FULL DUPLEX and ASYMMETRIC DUPLEX used. FULL DUPLEX means that transmission can take place both ways simultaneously. In ASYMMETRIC DUPLEX working, only one device at a time can transmit at full speed but the other can transmit simultaneously at a much lower speed. This allows for, say, one system to transmit a file at high speed, but the other to send commands controlling the transfer, asking to stop the transmission part way through the file, calling for another transfer and so on. You may sometimes hear the term HALF DUPLEX used. It means that only one device can transmit at a time, though each is capable of both transmitting and receiving in turn.

Note:

Unfortunately the term 'MODE' is also used in another sense, when we say whether the modem is set to respond to commands from the host PC or to pass on data to the telephone line. We say the modem is in LOCAL (or COMMAND) mode when it is set to obey commands from the host and to ON-LINE (or DATA) mode when it is set to transmit all input down the telephone line. ON-LINE and LOCAL are the terms used by MIRROR II, and displayed on MIRROR II's status screen, so we have used them here. You may see the terms COMMAND and DATA mode used instead, in other technical literature. (Here we are using 'host' in the sense of the PC to which the modem is attached).

f. Dealing with errors

A number of ways exist to check that the characters transmitted have been received correctly. We can employ the normal computer technique of using a parity bit with the seven bit code representing the character to ensure that all characters contain either an odd or an even number of bits. It doesn't matter which, as long as we agree on one option and stick to it for that transfer. The receiving end checks that all characters have correct parity. Of course, if an even number of bits is lost we can't tell. And even if an error is detected, we will probably have to transmit the whole message again. So other methods have been developed.

One other technique is to make up a 'packet' of characters, count all the bits in it (called a checksum) then send the packet and the checksum. The bits are counted again at the other end and compared with the checksum received. There are more sophisticated ways of counting the bits; you may come across the terms Blocksum and Cyclic Redundancy Checking (CRC), which are two such techniques. When whole files or sequences of files are to be transferred, error checking routines are used in conjunction with file handling software to form PROTOCOLS. We will look at those briefly next.

g. Protocols

Protocols are routines for transferring large quantities of data accurately and efficiently. They combine methods of breaking down files into manageable proportions with means of controlling the transfer, checking for errors and re-transmitting faulty blocks of data. Both the transmitting and the receiving systems must support the chosen protocol; very simple commands are then used to initiate the entire transfer.

h. Commands

We shall be using the word COMMAND frequently in this manual. This is a term which is used widely; operating systems, such as MS-DOS and PC-DOS each have a command set, MIRROR II has its own command set and in appendix H of this manual you will find the HAYES-compatible command set which is recognised by the modem.

When you give MIRROR II a command to alter any of the modem settings, entering a command from the MIRROR II command set, the MIRROR II software will send the appropriate HAYES-compatible commands to the modem. If you wish to use software other than MIRROR II to control your modem, you must ensure that it will issue the correct commands from the PPC modem command set, or will allow you to enter those commands yourself.

Once the modem is set into ON-LINE mode it will pass anything which is sent to it straight on to the telephone line, with the sole exception of the 'escape sequence', which we will describe later. If you are communicating with a program which recognises MIRROR commands, such as another MIRROR program or CROSSTALK, you may be able to control the other system by sending MIRROR II commands to it. Whether you can do so will depend on how the other system is programmed to respond to your input.

When the PPC modem is set to LOCAL (or COMMAND) mode, it is controlled by the commands sent to it by the host computer, that is, the computer to which it is fitted. When it receives a command to go into ON-LINE mode, it expects anything which is sent to it subsequently by the host to be data for transmission to the remote system. It does, however, recognise a unique sequence of characters as the 'escape sequence'. Sending the 'escape sequence' allows you to regain control of the modem at any time, for example, when incoming data could be seen to be corrupted and you wished to terminate the transfer.

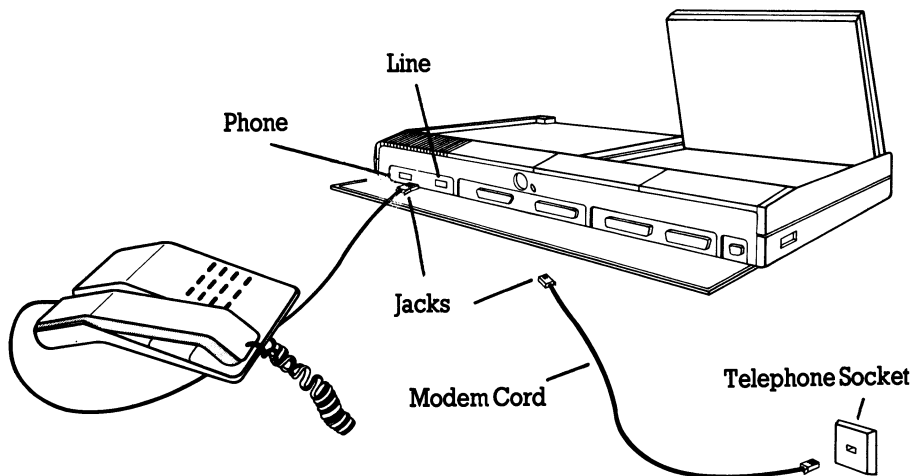
MIRROR II provides the 'BYE' command, which sends the necessary instructions to the modem to terminate a call. The 'AT' command (commonly allocated to the [Esc] key) simply allows you to control MIRROR, without terminating a call in progress.

The way in which the actual escape command is set and used in the modem is described at the end of the explanation of modem commands, in appendix H, section H2.

All the modem commands have been developed for a purpose, but it is unlikely that you will ever need all of them. The simpler ones are largely self-explanatory; the purpose of the others will be better understood when you have developed your skills to the point where you have a need for them. Don't worry if they seem obscure at first.

Do remember that even experienced professionals often need to use much care and patience in setting up a new communications link. By using the assistance provided in MIRROR II and studying the way in which the SCRIPT files use commands, in conjunction with these notes, you will soon become familiar with the basic essentials.

3.3 General details of the PC2000 modem



The AMSTRAD PPC modem is designed as an integral part of the PPC. There is nothing for the user to connect up, except for the phone line which is unplugged from its socket and plugged into the socket labelled "PHONE" at the rear of the PPC. Using the telephone extension cord provided, complete the telephone connection by plugging one jack into the socket labelled "LINE" at the rear of the PPC and the other into the telephone wall socket (this prevents loss of normal phone use).

Call monitoring is provided via the loudspeaker contained in the PPC, so although it is normally convenient to connect the telephone to the modem, it is not essential to do so. The PPC has an intelligent modem, with Auto-dialling and Auto-answering capability.

Technical information relating to the circuitry or construction of the modem cannot be supplied to the user.

The modem is intended for use under the control of the MIRROR II software, developed by Softklone Distributing Corporation, but uses a HAYES-compatible command set and should respond satisfactorily to any communications program which uses those commands.

The modem is compatible with modems operating at speeds of 75 to 2400 bps Bell 103, 212A and CCITT V22bis modes. Details of these modes, which are prescribed in the relevant Bell and CCITT standards, are given in appendix H, section H.5.

The full HAYES-compatible command set for the modem is given in appendix H, section H.2.

The special registers, controlled by the user's software, are listed in appendix H, section H.3, and the Result Codes in section H.4.

Before attaching any equipment to the public telephone system you should read the information in appendix H, section H.1.

CAUTION

There are legal restrictions on the importation and use of computers and modems, even for a short period, which vary widely between different countries. Users intending to take a computer or modem abroad should be careful to learn and comply with the requirements of Customs and Telephone authorities in the countries to be visited. In some cases a substantial bond must be deposited, which will be retained by Customs until the equipment is returned to its country of origin. Before leaving they must also ensure that they have the necessary documentation to avoid liability for duty on their return. Regulations are always liable to change, so please ensure that you have the latest information when planning to take equipment abroad.

It is illegal to connect to any public telephone network a modem of a type not approved for such connection by the relevant authority.

3.4 Introducing MIRROR II

MIRROR II is designed to run on PC compatible systems under the control of the MS-DOS or PC-DOS operating systems. Before using MIRROR II please make a copy of the disk supplied with your AMSTRAD PPC. Use the copy for your normal work and keep the original disk in a safe place, away from excessive heat or strong magnetic fields, in case your working copy is damaged and you need to take a further copy. It is permissible to make a copy for your own use, on one system only, but you must not allow it to be used on another system except as permitted in the licence agreement printed in this manual.

MIRROR II has the capability to perform most of the PC communications operations currently possible over the public telephone network, in an efficient and user-friendly manner. It provides all the facilities of the well-established CROSSTALK XVI communications program, plus a number of valuable enhancements including additional protocols, built-in text editor and the capability of running in the background whilst other software is running. Terminal emulation and password protection features are included.

A MIRROR II script file, a sequence of MIRROR II commands which can be loaded and run by entering a single identifier at the keyboard, can be set up to cause the modem to:

Dial a number, or answer an incoming call.

Select a transmission speed (explained later in this section).

Send passwords and identification.

Transmit or receive data.

Monitor incoming data for specific codes and respond with further commands.

Terminate the call on completion.

Though other communications software may be used, particularly if designed for use with HAYES compatible modems, (or you may of course develop your own programs when you have the necessary knowledge), we have chosen MIRROR II because it allows you, very easily, to exploit the features of your PPC modem to the full, whilst offering a very simple approach to communications for the beginner.

MIRROR II provides commands to allow you to use the protocols in common use which you are most likely to encounter. The system with which you intend to communicate must also be able to support the protocol which you intend to use. Communicating with other MIRROR II or CROSSTALK systems should present little difficulty; however, when communicating with other software packages you may sometimes need to obtain details of precisely how they implement the protocols.

The protocols supported by MIRROR II are:

HAYES; CROSSTALK; XMODEM (single and multi-file, with optional checksum or CRC error checking; YMODEM; KERMIT.

That doesn't mean you can't use any other protocol you wish, only that MIRROR II has commands provided specifically to use those protocols with a minimum of effort on your part. For example, suppose you wish to transmit the file named report.one, on your drive A: to drive C: on a system whose 'phone number is 1-404-555-1212 and which communicates at 1200bps. The remote system supports the CROSSTALK protocol, for which MIRROR II provides the XM command. You would load MIRROR II and use the commands:

MO call	to set your system into call mode.
SP1200	to set the speed.
NU 1-404-555-1212	followed by
GO	to dial the number.
XM a:report.one c:	to transfer file report.one from your disk a: to the remote disk c:.
BYe	to terminate the call.

You could type in each command in turn, in reply to the 'command?' prompt at the bottom of the MIRROR II screen. Alternatively, you could create a 'script' file, using MIRROR II's built-in editor and 'script' programming language, but then you would need to include some more commands so that, for example, MIRROR II didn't try to start the transfer before the call was connected.

During the transfer MIRROR II provides a display telling you how the transfer is progressing, what percentage of the file has been sent and how many errors have occurred.

As you see, very few commands are needed; though if you are going to use the same routine regularly it is worth the extra effort of creating a script file. If you wish, MIRROR II will automatically save the settings you have used for call, on a file which you can simply load next time you want to use the same settings.

All the MIRROR commands are given in this manual, together with a description of their use. For a more detailed description of MIRROR II we recommend that you obtain the full MIRROR II manual, published by Softklone Distributing Corporation, from your dealer.

Note:

It is very easy to change any of the parameters set up by MIRROR II's command files, by simply entering the relevant commands when in LOCAL mode. If the SAVE command is then used, the original settings IN THE COMMAND FILE will be overwritten. It is also very easy to modify the script files, using the built-in text editor. Once again, if the save option is then used the original files will be overwritten with the new data. It is quite safe to experiment with parameter changes and to read or even modify files, using the text editor, provided that the modified files are simply abandoned if the changes are not required. Pressing the key for the 'AT' command (usually [Esc]) when in LOCAL mode allows you to exit safely and simply from any file, without overwriting the original.

Use particular care if you decide to change the keys used for the 'AT' or 'SW' functions. It is very easy to do and can be very confusing if you forget later on and can't seem to regain command of the program.

MIRROR II has a very comprehensive HELP facility available on-line. Simply type 'help' at the Command ? prompt. For help on a specific command, type the command followed by a '?' .

3.5 NEWUSER procedure

Select the NEWUSER command file, if you wish MIRROR to assist you with automatic logon procedures for many of the major information utilities. NEWUSER will list a menu and prompt you to enter a letter corresponding to the service you wish to call. For example, if you choose COMPUSERVE, NEWUSER will prompt you for the telephone number, modem speed, and your COMPUSERVE account number and password. NEWUSER saves this information automatically in a new command file called COMPUSRV, which will be listed as an 'available command file' on the Status Screen. At this point, you can call the service or select another 'available command file'. You can use NEWUSER to create command files for each of the services listed in NEWUSER's menu that you use.

3.6 General Calling Procedures

The fundamental steps are:

- a. Dial a number**
- b. Identify yourself**
- c. Say what you want to do**
- d. Send or receive files, from a database or bulletin board;
or**
- e. Interact with the host system, if acting as a terminal to
that system**
- f. Terminate the call**

There are two principal differences from making an ordinary speech call:

- i) Since there are a variety of transmission speeds in common use, you must select the speed used by the service which you are calling. The PPC modem will initially set itself to the speed of the host computer. When a connection is made, the speed of the incoming carrier signal is detected and the appropriate CONNECT code is returned, from your modem to MIRROR, when the speed of the incoming carrier is the same as that to which your modem is set. The carrier speed is specified in the code (for example, CONNECT 1200 for a 1200 baud carrier responding to a call at 1200 baud) except in the case of 300 baud, when the result code is simply CONNECT. A carrier at the wrong speed will not normally be detected, so the result code NO CARRIER will be returned. However, some standards permit the use of alternative speeds. In these cases the appropriate CONNECT code will be returned. For example, a transmission in CCITT V22 bis mode at 2400 baud, answered by a 1200 baud carrier, will result in the code CONNECT 1200 being returned. It is then necessary to change the speed of your transmission to that of the incoming signal, before attempting further communication. The MIRROR II Speed command includes an option whereby this will take place automatically.

- ii) You need to know what sort of inputs the service expects from you and whether you can receive data from it, send data to it, or 'converse' with it. It may transmit 'menus' to your screen, helping you to see what options are available to you, but the actions required of you will depend entirely on the way in which the software of the called system has been designed. There are a number of methods (called 'protocols') of transferring large files; breaking them down into manageable sections and performing checks for errors in the transmission. If you are planning to make such transfers you will need to ensure that both systems have the software necessary to support the protocol which you wish to use.

Before commencing the call therefore, it is necessary to obtain sufficient details of the service to know what speed to set your system for and what to do once you are connected. It will almost certainly be less costly to do this by post or an ordinary telephone call, than by experimenting with your system on-line.

Example of making a call, using MIRROR, following the above steps:

(Each command is completed by pressing Carriage Return)

Enter the commands

NU 1-404-555-1212

GO (to dial the number 1-404-555-1212 directly)

or

NU 9,1-404-555-1212

GO (to dial the same number via a PABX (internal) exchange requiring a 9 to obtain an outside line.)

After a short time you should hear the loudspeaker on the modem clicking as the number is dialled (or sounding the DTMF tone, if tone dialling). After dialling, the modem will wait for the period set in register S7 for an answer, then abandon the call if no carrier has been detected and send the result code NO CARRIER to your computer (see appendix H, section H.4, for result codes).

If the call is successful, a carrier will be detected within the waiting period and the modem will return the appropriate CONNECT code (see Appendix H, Section H4). (In the default mode the loudspeaker is disabled on detection of a carrier, however if the command ATM2 is received by the modem, the speaker will remain enabled and the carrier will be heard as a high-pitched note.)

If you have connected to a service such as Compuserve you will then see a message on your display identifying the service. Depending on the service you are calling, you may then be asked to enter a password and your name or identification number. Once these are correctly entered, instructions for continuing the transaction will normally be displayed.

When you have completed the call, the remote system will normally respond to a given command, for example OFF, and will drop its carrier, terminating the call. This will cause the return code NO CARRIER to be sent.

If the modem is connected to an information service such as Compuserve, the line may not be dropped for several minutes. In this case you would force the modem off-line by pressing the ESCape key and typing in the BYe command.

Notes:

- i) The modem powers up in LOCAL mode, ready to accept commands from your system. Once a call is connected, it is set to ON-LINE mode and whatever you input is sent to the remote system. Entering the 'escape sequence' will return control to you by resetting the modem to LOCAL mode.

- ii) Resetting the modem to LOCAL mode, using the 'escape sequence', does not terminate the call; further commands may be sent to the modem and the call resumed.

The escape sequence is described at the end of the description of modem commands, in appendix H, section H2.

3.7. WORKING WITH MIRROR II

3.7.1 Getting Started

- 3.7.1.1 Conventions**
- 3.7.1.2 Loading MIRROR**
- 3.7.1.3 Using MIRROR for the First Time**
- 3.7.1.4 MIRROR Command Usage**
- 3.7.1.5 On-line Help**
- 3.7.1.6 Making a Call Using MIRROR**
- 3.7.1.7 The Terminal Screen**
- 3.7.1.8 Trouble Shooting**
- 3.7.1.9 Putting MIRROR into the Background**
- 3.7.1.10 Terminating a MIRROR Session**

3.7.2 Status Screen

- 3.7.2.1 Sample Status Screen**
- 3.7.2.2 Command Line Editing**
- 3.7.2.3 Parameter Tables**
- 3.7.2.4 The Window**

3.7.3 Communications Control Features

- 3.7.3.1 Speed (Baud Rate)**
- 3.7.3.2 Telephone Number**
- 3.7.3.3 Serial Port**
- 3.7.3.4 Call or Answer Mode**
- 3.7.3.5 Automatic Redial**
- 3.7.3.6 Duplex Mode / Data Bits / Parity**
- 3.7.3.7 Stop Bits**
- 3.7.3.8 Call Progress Monitoring**

3.7.4 Capturing Data or Files

- 3.7.4.1 Capture to Disk**
- 3.7.4.2 Capture to Memory**
- 3.7.4.3 Interrupting Data Transmission**
- 3.7.4.4 Character Filtering**
- 3.7.4.5 Retro-Capture**
- 3.7.4.6 Browsing Captured Data**

3.7.5 Sending Data or Files

3.7.5.1 Send Command

3.7.5.2 Character Filtering

3.7.5.3 Flow Control

3.7.5.4 Communicating with a Prompting System

3.7.5.5 Communicating with Slower Systems

3.7.6 File Transfer Using Protocols

3.7.6.1 Introduction

3.7.6.2 Using the Crosstalk Protocol

3.7.6.3 XMODEM (Single File) Protocol

3.7.6.4 XMODEM (Multi-File) Protocol

3.7.6.5 KERMIT Protocol

3.7.6.6 Hayes Protocol

3.7.6.7 YMODEM Protocol

3.7.6.8 Examples

3.7.7 Advanced Features

3.7.7.1 Background Use of MIRROR

3.7.7.2 Terminal Emulation

3.7.7.3 Saving / Reloading / Modifying Command Files

3.7.7.4 Learn Command

3.7.7.5 Integrated Text Editor

3.7.7.6 Programming with Script

3.7.7.7 Command Completion Status

3.7.7.8 Transaction Log

3.7.7.9 Answer Mode

3.7.7.10 Password Maintenance Utility

3.7.7.11 Function Key Usage

3.7.7.12 Remote Command Capability

3.7.7.13 Operating System-like Commands

3.7.7.14 Screen Command

3.7.7.15 Setting the Special Keys

3.7.1 Getting started

3.7.1.1 Conventions

In this manual we will be presenting a series of examples involving the use of the computer. In these examples, we have adopted the following notation:

- a. Return or Enter are synonymous with [↵]
- b. The character ^ before a letter means to press and hold the CONTROL [Ctrl] key while typing the letter that follows. For example, ^C means to hold down the CONTROL key while simultaneously pressing the letter C.
- c. y means to press the Y key (either upper or lower case).
- d. This style of **type** represents input required by you.
- e. The abbreviation 'afn' is for ambiguous file name, for example, *.doc or x?.exe.
- f. The abbreviation 'ufn' is for unambiguous file name, for example, manual.doc.

3.7.1.2 Loading MIRROR

- a. Turn on your computer and load your operating system (DOS 2.0 or later).
- b. Put the MIRROR working disk in drive A.
- c. At the A> prompt, type

```
mirror [↵]
```

- d. You should now see the MIRROR banner. Press any key to display MIRROR's Status Screen.

3.7.1.3 Using MIRROR for the first time

When you enter MIRROR you may be offered the choice of several Script files which you may run to select a particular service, customize files with your personal parameters, obtain general help, or change the display colors. You may bypass any of these Scripts by pressing the [Esc] key twice; once to interrupt script processing and a second time to enter command mode.

Mirror has three screens available for display:

- a. The Status Screen, which displays the current settings of the primary MIRROR parameters, as set up by the last script file which was loaded.

- b. The Terminal screen, which is used to display data transferred via the modem. (The **[Home]** key will switch between these two screens).
- c. The Alternate screen, which can be selected by the user's Script files (or a directly entered `sc a` command) and used for any desired purpose.

When the `Lo d` command is entered, the titles of all available Command (type `. XTK`) files are displayed, with selection numbers, on whichever screen is in use at that time. On the Status screen they are displayed in the 'Window' at the lower part of the screen.

The title 'Command File Directory' is shown and the message:

```
Enter number for file to use ( 1- n ):
```

is displayed at the bottom of the screen.

When you enter a number and press **[↵]** that file will be loaded and run.

There is a second group of Script files (type `. XTS`) which are normally run by being called from a Command file, using the `D0` command. Entering the `D0` command on the command line will cause the directory of these files to be displayed UNLESS the execution of a previous file has been suspended, by use of the **[Esc]** key, in which case execution of the suspended file will be resumed instead. In this case, entering the `Lo d` command, cancelling it with the **[Esc]** key without selecting a number and then entering the `D0` command, will cause the Script file directory to be displayed.

Selecting the appropriate number will cause one of these files to load and run, but if it is a file intended to be called by a Command file or another Script file, it may not function correctly.

The `HELP` command file allows you to receive general help (as opposed to specific help on an individual command) on various topics. `HELP` is initiated either by typing `lo help` or `he ??` followed by a Return when the `Command?` prompt is displayed at the bottom left-hand corner of the screen.

You can use the `Le a Rn` command to create your logon procedure. Refer to the on-line `HElp` information for this command. Remember that if you need more information on any of the features that are described in this Reference Guide, use the `HElp` screen for the command of interest or the general `HELP` system (`he ??`) or you may want to purchase the companion User's Guide which goes into much more depth on all the features of `MIRROR`.

3.7.1.4 MIRROR command usage

The Command Line, the bottom line of the Status Screen, is your principal interface with MIRROR. You can enter MIRROR commands any time the Command? prompt is displayed at the bottom of the screen. If you are in the Terminal Screen and need to bring up the Command? prompt, press MIRROR's Attention key, which is initially set to the [Esc] key.

To enter a command, type the first two characters of the command, followed by its parameters (if any), followed by a RETURN. If parameters are required for the command and you do not enter them, you will be prompted for them. Enter commands in either upper or lower case. In addition, there must be at least one space between a command and its first parameter. For example, to use the on-screen help information for any MIRROR command, enter

```
he [↵]
```

for the Help command. You will see the first of a series of screens containing all MIRROR's commands. To get help for the Speed command, you could enter

```
he sp [↵] or sp ? [↵]
```

and the help information for the Speed command would appear on the screen.

Several MIRROR commands toggle simple ON/OFF switches, and are called toggle commands. When a toggle command is entered without an argument, MIRROR displays the switch setting on the command line and asks if you want to change it. A toggle command can also be entered with an argument. For example, you can turn Off the Capture mode by typing

```
ca off [↵] or ca - [↵] or ca 0 [↵]
```

and turn it back On again by typing:

```
ca on [↵] or ca + [↵] or ca 1 [↵]
```

You can also toggle a switch to its opposite setting by using the slash (/) character as an argument to the command. For example, if Capture mode is in the On setting, you can toggle it Off by typing:

```
ca / [↵]
```

Follow this procedure for any of MIRROR's toggle commands.

If you need to select an 'available command file' when the Command? prompt is displayed, enter the Load command:

```
lo [↵]
```

MIRROR will then prompt you to select a file from the 'Command File Directory'.

3.7.1.5 On-line help

On-line help is available through the `HEl p` command and Help System. Anytime you enter a command that requires parameters and you need help remembering what information is required by the command, enter a question mark (?) or press the Function Key 1 [**F1**], and MIRROR will display an explanatory message. For example, typing

```
l r ? [↵]
```

will generate the HELP display for the `LeaRn` command, as would typing `l r` and pressing the [**F1**] key. You should use the on-line HELP any time you are uncertain how a command works or what its parameters are.

3.7.1.6 Making a call using MIRROR

When calling another computer, your microcomputer should be in `Call M0de`. If another computer is going to be calling you, you will have to be in `Answer M0de`. You can change the mode by using the `M0de` command. The current `M0de` setting is displayed under the heading 'Connection Parameters' on the Status Screen.

You can make a call using MIRROR in two ways, by entering commands or by using existing command files.

You can make a call manually using commands. Use the `NUmber` command to enter the phone number for the computer you want to call. An example is:

```
nu 1-404-555-1212 [↵]
```

If the phone number field is blank, the dialing procedure is skipped. This string can be up to forty characters in length. The default is a blank phone number.

You can logon automatically by using an existing command file or one created through the `LeaRn` command. At the command prompt simply `L0ad` the file you wish to use. The dialing process is initiated automatically.

Use the `SAve` command to save your parameters to a command file (named by you). For instance, typing

```
sa callu [↵]
```

on the command line will save all current MIRROR parameters to the file `CALLU`. From now on you can set your parameters by simply using the `L0ad` command to 'load' the command file, `CALLU`, into the computer. Also, `CALLU` would now be a choice in the 'available command files' list which is displayed on the Status Screen at start-up time. `CALLU` could then be loaded by typing the number representing it followed by a Return. Finally, use the `G0` command to initiate the call.

After you have successfully logged on to another computer, you should see the message in the top right-hand corner of the Status Screen change from Off-line to On-line. You should also see a sign-on message from the remote computer. You will now be in terminal mode and should see MIRROR's Terminal Screen.

You can receive (download) a file from the other computer using the `Capture` command. You can also transmit files using verification protocols with error checking and recovery such as Crosstalk, Hayes, KERMIT, single and multi-file XMODEM, and YMODEM.

3.7.1.7 The terminal screen

MIRROR displays the Terminal Screen when it is acting as a terminal into another computer. This screen fills when you type something onto it or when it displays the output of a communications session. Note that MIRROR will not pause as the Terminal Screen fills up. Press `^S` to temporarily suspend the displaying of lines. Press `^Q` to restart the process.

While your computer is communicating with another system (terminal mode), the command line on the Terminal Screen will look something like this:

```
Esc for Command?, Home for Status | Capture Off | On: 00:31:06
```

The first part of the command line displays the keys to get MIRROR's attention or to switch back to the Status Screen. To enter MIRROR commands from the Terminal Screen, press MIRROR's **A**Ttention [**Esc**] key to bring up the Command? prompt. Use the **S**Witc**H** [**Home**] key to switch back and forth between the Status and Terminal Screens.

The second part of the Terminal Screen's command line tells you the status of the `Capture` mode, which can be On, Off, Paused or currently capturing data to a disk file.

The third part of the command line is the `T`imer display. While the `T`imer switch is On, MIRROR will display the elapsed time (in hours, minutes, seconds) since the carrier signal was detected from the other system. If the timer is Off, the message On-line is displayed.

3.7.1.8 Trouble shooting

If you have experienced problems making a call to another computer or logging on to another computer, there are several things to check to remedy the problem(s). If you did not see a logon message from the computer you were trying to call, the most likely problem is the baud rate that you specified. You will need to check some of the MIRROR communications parameters to see whether one or more do not match with what the other system is expecting. Press the `S`Witc**H** key (the [**Home**] key is the default) to return to the Status Screen and command mode. Now check the following parameters, in order, to see if you can find the problem. If you find a problem with any of the parameters listed below, change it and type `go [↵]` again to see if you can complete the connection.

- a. Is the serial POrt correct? Is your modem connected to the POrt number displayed on the Status Screen? (The PPC internal modem is connected to POrt 2.)
- b. Is your machine in Call MOde? If the other computer is a microcomputer, it should be in Answer MOde.
- c. Is the SPeed (baud rate) on your machine the same as that expected by the computer you are calling? If you are using the prefix 'a' on the baud rate, try it without the 'a' because you may be using Call Progress Monitoring incorrectly.
- d. Are the DUplex, DAta bits and PArity parameters set to those required by the system you are calling?

3.7.1.9 Putting MIRROR into the background

You can put MIRROR into the background to run another program while continuing the current call and activity of MIRROR by simultaneously pressing the left and right Shift keys. To return MIRROR to the foreground, simultaneously press the left and right Shift keys again. You should make sure that this key combination (left and right shift keys) does not conflict with any other memory resident programs you are using. If so, you will need to change the key settings of one of the programs.

3.7.1.10 Terminating a MIRROR session

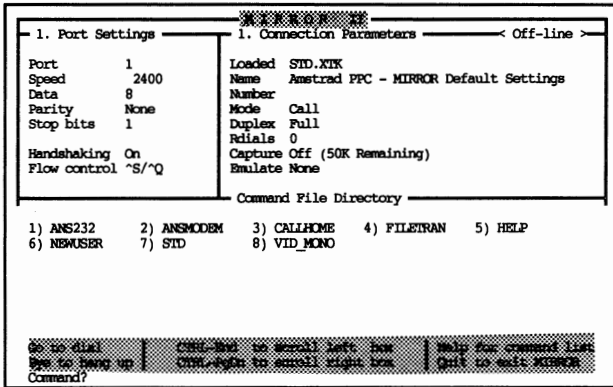
There are two ways to terminate a MIRROR session:

- a. Use the BYe command to terminate a call in progress and remain in MIRROR.
- b. Use the QUit command to terminate a call in progress and return to the operating system.

3.7.2 Status screen

3.7.2.1 Sample Status screen

The first thing MIRROR displays (after its banner) is the Status Screen. The on-line/off-line status indicator in the upper right-hand corner of the screen displays the current communications status (On-line, Off-line or Local) for the session. The top section of this screen contains the parameter tables. The bottom portion of the screen has a multipurpose Window for information display. Finally, the last line of the Status Screen is the command line. A sample of a Status Screen at the start of a session follows:



3.7.2.2 Command Line editing

Since typing errors can occur while entering commands on the Command Line, MIRROR's Command Line editing feature permits you to correct errors without retyping the entire entry. If you do not enter the command's argument for commands that have character strings as arguments such as `NUMBER`, MIRROR will display its usual prompt and display the current parameter setting for you on the command line. You can edit the command line according to the following procedure:

- The cursor is positioned at the first character of the setting and you are in overtype mode. That is, you will be able to edit the setting by overtyping the displayed characters.
- To move the cursor left or right one character at a time, without disturbing the characters displayed, press the left or right arrow key. To move the cursor left (right) a word at a time, press the left (right) arrow key and the **[Ctrl]** key simultaneously.
- To insert one or more characters at any place in the setting, press the **[Ins]** key. This will put you into insert mode and the characters `Ins` will be displayed in the rightmost position on the command line. To return to overtype mode (the default mode), press the **[Ins]** key again. MIRROR will automatically return to overtype mode when you press Return to enter the new setting.

- d. To delete the character on which the cursor rests, press the **[Del]** key. To delete the character to the left of the cursor, press the backspace key. To delete all characters starting with the current position of the cursor and continuing through the end of the line, press the page down **[PgDn]** key.
- e. The following is a summary of MIRROR's command line editing keys:

INSert	Toggle overtype/insert modes.
Left arrow	Move cursor left one character.
Right arrow	Move cursor right one character.
Ctrl-left arrow	Move cursor left one word.
Ctrl-right arrow	Move cursor right one word.
Delete	Delete the current character.
Backspace	Delete the character to the left.
Page Down	Delete to end-of-line.

For example, enter the `NUMBER` command and MIRROR will display the current phone number, such as:

```
Phone Number (enter/exit): 1-404-555-2121
```

Note:

The cursor is positioned at the first character in the phone number. Suppose you forgot to put in the area code 1-. Press the **[Ins]** key to turn on insert mode. The command line now looks like this:

```
Phone Number (enter/exit): 1-404-555-1212      Ins
```

Now insert the area code by typing 1-. The command line will display:

```
Phone Number (enter/exit): 1-404-555-1212      Ins
```

Press **[↵]** to complete the editing and enter the new phone number. You will also be returned to overtype mode.

3.7.2.3 Parameter tables

MIRROR groups its most commonly used parameters into tables on the Status Screen. The parameter names are listed along with their current settings. Remember, only the first two characters of the command need be used to specify the command to change the setting you see displayed. A brief discussion of the parameter tables follows. Their format consists of a heading under which the parameters and their current settings (if any have been assigned) are listed. A number precedes each table for easy reference.

Parameter tables appearing in the left-hand box are:

1. The Port Settings table occupies the left box when the Status Screen first comes into view. The communications **PORt**, transmission **SPeEd**, **DAta** bits, and **PARiTy** are settings necessary to almost any communications session. The table is given prominence for this reason. It also contains the **FLow** control settings.
2. The table of Security Features contains the settings with which you can control access to your system by remote users.
3. The Trigger Characters table lists the keys that perform special functions when pressed during the operation of **MIRROR**. The **ATtention** key, in terminal mode, gets **MIRROR**'s attention, so that you can type a local command. The **COmmand** key, on the other hand, permits you to specify a key that can be typed by a user on another system who has dialed into your system. By pressing this key, the remote user can order **MIRROR** to execute a command on your system (your system must be in **Answer MOde**). The **SWiTch** key toggles between the Status Screen and the Terminal Screen.
4. The Send Options table contains the parameter settings used with the **SEnd** command to transfer data to another computer.
5. The Display Modes table lists the parameters that govern the operation of your display screen. Included in this list are the commands that control screen color and terminal emulation.
6. The final table displayed in the left box lists parameters governing **MIRROR**'s protocol Transfer Options. Prominently displayed are the settings used in data transfer via the Kermit protocol.

Parameter tables appearing in the right-hand box are:

1. The Connection Parameters table occupies the right box when the Status Screen is first displayed. This table lists parameters important to the establishment of a connection with a computer, including the telephone **NUmber** to be dialed.
2. The settings of the ten Normal function keys.
3. The settings of the ten Shifted function keys.
4. The settings of the ten Control function keys.
5. The settings of the ten Alternate function keys.
6. The System Settings table; displays disk drive, subdirectory, and path information.
7. The Modem Strings table displays the character string parameters that are sent to your modem when you enter the **G0** command. This table also contains the parameter setting that governs **MIRROR**'s call progress monitoring feature.
8. The Receive Parameters table displays the settings that govern the handling of data received over communications lines by **MIRROR**. **Data CAPture**, filtering, and printing controls are all listed.

3.7.2.4 The Window

MIRROR provides you with a nine line Window through which to view both the 'Filter Table' and the results of certain commands. If the information you are viewing is longer than nine lines, MIRROR allows you to browse through it, one frame at a time, by pressing Return. You may only browse in one direction: downward. If you wish to cancel browse mode (for example, when viewing a lengthy file using the `TYPE` command), press the `ATTENTION` key.

The auxiliary parameter information that can be displayed in the Window is the 'Filter Table'. This table specifies which control characters MIRROR will filter out when the `INFilter` switch is set to On. The `Filter` command brings this list of control characters into the Window. Once they are in the Window, they remain until you `Clear` the Window or use another command that overwrites the list.

Note the two lines just beneath the nine line Window and immediately above the command line. These lines are broken into three segments. The middle segment tells you how to scroll the left and right boxes at the top of the Status Screen. The first and third segments tell you the commands used to get MIRROR to dial a telephone number (`GO`), to terminate a call (`BYE`), to get `HELP`, and to exit MIRROR (`QUIT`).

MIRROR also has other uses for the Window. The following is a compilation of the more important commands that use the Window to display their output.

- a. It is possible for you to save all parameters for a given session on a command file. A 'Command File Directory' of the currently logged drive will be displayed when the `Load` command is entered without an argument. MIRROR also displays this list in the Window at the beginning of each MIRROR session.
- b. A 'Script File Directory' can be displayed using the `D0` command. Script files are similar to command files but may also include program logic and branching statements.
- c. The `Dir` command displays disk directories.
- d. You can browse through the contents of either a disk file or the capture buffer using the `TYPE` command.

During the transfer of files using any of MIRROR's verification protocols, MIRROR uses the Window to display a progress report on the status of the file transmission.

3.7.3 Communications Control Features

The Status Screen displays the preset values of many of the modem and other communications control parameters. Generally, these preset values will work in most cases. If you change the value of any of these parameters, the value is immediately updated on the Status Screen. The other modem control parameters, not initially displayed on the Status Screen, may be displayed in the boxes at the top of the Status Screen by using the scrolling commands or by changing one of these parameters.

All the commands used to change the communications control parameters will prompt you for input if a required parameter is not given along with the command. Any parameter changes you make will stay in effect only until you exit to the operating system, unless you use the `SAVe` command to preserve the changes in a command file.

3.7.3.1 Speed (Baud Rate)

The `SPeEd` command allows you to select a new baud rate. You will be prompted as follows:

```
0110, 300, 600, 1200, 2400, 4800, 9600, 19200, etc. baud?
```

Only the first two digits need to be entered to correctly specify the baud rate (for example, 24 for 2400). The baud rate default value is 1200.

3.7.3.2 Telephone Number

The `NUmber` command allows you to specify the phone number of the remote site you want to dial using `MIRROR`. Consult Appendix H, section H2, for legal characters which can be used. `MIRROR` ignores leading blanks when you enter a phone number. If the phone number field is blank, `MIRROR` skips the dialing procedure. To enter a blank phone number, press Return when prompted to enter the phone number. After the previously entered phone number appears, delete it using the `[PgDn]` key and enter the blank number by pressing Return. The default value is a blank phone number.

3.7.3.3 Serial Port

The `POrt` command specifies the serial port on your microcomputer that `MIRROR` will use for communications with another computer. The serial ports are typically numbered from 1 up to 4. The PPC uses port 1 for the external serial connector (for serial printers etc). The PPC internal modem is connected to Port 2.

3.7.3.4 Call or Answer mode

The `MOde` allows you to select the 'role' that `MIRROR` will play in a connection. The choices are Call and Answer. If you are dialing into another system, you should select the Call role. If someone else is going to be dialing into your system, you should select the Answer role. The default value is Call.

3.7.3.5 Automatic Redial

The `RDial` command specifies the number of times an auto-dial modem will automatically redial the `NUmber` if the connection is not completed in the first dialing. Zero means do not redial if the initial call is unsuccessful. The value can be set from 0 to 255 to indicate the number of desired retries before giving up.

3.7.3.6 Duplex mode/Data bits/Parity

An important point to realize is that the duplex, data bits, and parity values you specify need to be the same as those expected by the computer you are dialing. Before changing these, try running with the default settings.

3.7.3.7 Stop bits

The `STop` command allows you to select the number of stop bits on each character transmitted. You will probably never need to change this parameter from its default value of 1.

3.7.3.8 Call Progress monitoring

`MIRROR` can perform call progress monitoring. `MIRROR` can monitor the progress of your modem while it is attempting to dial another system if your modem has this capability. `MIRROR` can detect a busy signal and terminate the current dial sequence (immediately beginning the 'Waiting to re-dial' period, if `RDialing` has been requested), without continuing to wait for a connection that will not happen on this dial. `MIRROR` interprets the result codes sent to it by your modem during the dial process. `MIRROR` performs call progress monitoring any time you have selected automatic baud rate adjustment using the `SPEED` command (you do this by prefixing one of the standard `SPeed` settings with the letter `a`; for example: `a2400`).

3.7.4 Capturing Data or Files

You can now use `MIRROR` to make your computer communicate with a mainframe or another microcomputer, with your system playing the part of a computer terminal. `MIRROR` also permits you to transmit entire files of data from your system to another, something that will be discussed in the next two chapters. What is probably more important to you, however, is `MIRROR`'s ability to retain or capture a copy of the data you receive from another system.

`MIRROR`'s data `Capture` features allow you to capture data to your computer's memory or directly to disk. `MIRROR` also permits you to weed out extraneous control characters that might interfere with data transmission. Then, once you have captured your data, `MIRROR` lets you permanently save it, and provides several ways for you to browse through it.

You can use `MIRROR`'s retro-capture feature to bring back into the capture buffer some of the information that has scrolled off your screen. `MIRROR` also gives you the ability to save a single screen's worth of information. A permanent copy can be made on disk using the `PIcture` command, while the `SNAPSHOT` command permits you to make a copy that will last only until you leave `MIRROR`. The `PRinter` command permits you to keep a hard copy transcript of your session.

3.7.4.1 Capture to disk

To capture data to a disk file, simply supply the file name as the argument to the `CAPture` command. You may enter this command either before or after you begin communicating with another system. Remember that once you have started communications, you will have to press the `ATTention` key to get `MIRROR`'s attention before entering your `MIRROR` command.

The following example shows how to direct `MIRROR` to write captured data to the file `INDATA.DOC` on the 'B' drive:

```
ca b:indata.doc [↵]
```

When you know ahead of time that you intend to add to, or overwrite, an existing file, you may enter this information as a second argument to the `CAPture` command. The additional argument is `/a` when you intend to add data to a file, and `/e` when you want to erase the file before adding to it (overwrite). Note that it is not necessary to separate either the `/a` or the `/e` from the file name parameter. However, you may not insert a space between the slash (`/`) and the 'a' or the 'e'.

You are now ready to begin capturing data. Start communications by typing `go [↵]`. Typing `ca / [↵]` will suspend capture mode and `MIRROR` will display the word `Paused` on the Status Screen beside the `CAPture` parameter. Typing `ca / [↵]` will turn capture mode back on.

When you have finished capturing data, turn the capture mode off with

```
ca off [↵]
```

to close the capture file. When choosing disk file names, exercise caution: avoid using file names that begin with the 'at sign' (`@`) or the less than sign (`<`).

3.7.4.2 Capture to memory

The capture to memory function operates in much the same way as capture to disk. However, different command parameters or arguments are specified. When you are ready to begin capturing data to memory, type

```
ca on [↵]
```

Capture mode can be `Paused` or `toggled Off` as explained in the previous section.

The main difference between this procedure and capturing data to disk is that data is captured to your computer's memory. This memory area is called the capture buffer. By default, `MIRROR` sets aside up to 50K of memory for the capture buffer. If there is not 50K of memory remaining on your system (after `MIRROR` and your operating system are taken into account), `MIRROR` limits the size of the capture buffer to the amount of memory that is still available. Within your system's memory constraints, `MIRROR` permits you to adjust the upper limit on the size of the capture buffer. To change the capture buffer limit, use the `Size` option to the `CAPture` command.

Having turned capture mode On, you are now ready to begin capturing data. Start or resume communications as shown in the previous section. You will see

```
Esc for Command?, Home for Status || Capture On, nnK free
```

displayed on the command line. This tells you how much free space (in terms of thousand characters) remains in the capture buffer. As the capture buffer fills, this line will be updated to let you know how much unused space remains.

When the capture buffer is full, MIRROR issues a final warning and turns the capture mode Off. Your screen will continue to display any data that is being sent. However, this data will not be written to the capture buffer, which is now full. Therefore, this new data will be lost (once the retro-capture buffer fills).

WRITE COMMAND:

You can use the `WRite` command to write the contents of the capture buffer to disk. Before this can be done, you must temporarily interrupt data transmission from the other system by pressing the 'STOP transmission' character that is recognized by the other system. This is usually `^S` (review the `FLow control HElp` screen for a more complete discussion). Once data transmission has been interrupted, press the `ATtention` key, MIRROR will display the prompt `Command?` on the command line.

Once you enter the `WRite` command, MIRROR writes the capture buffer to the file specified and erases the capture buffer. You may then resume data transfer by pressing the 'RESTART transmission' character, which is normally `^Q`.

3.7.4.3 Interrupting data transmission

If you will be communicating with a system that does not follow the `^S/^Q` convention, you need to be able to adjust the values of the `STOP` and `RESTART` transmission characters using the `FLow control` command.

3.7.4.4 Character filtering

MIRROR provides you with three commands that permit you to have data received by your system preprocessed before being displayed. These are `LFauto`, `DEbug`, and `INfilter`. The `INfilter` command is used in conjunction with another command, `FilteR`, which specifies the characters to be filtered.

3.7.4.5 Retro-capture

As incoming data is written to your screen, MIRROR is also writing it to a circular buffer that can hold 4096 characters. This is called the retro-capture buffer, which operates even if capture mode is Off. With retro-capture, you have the ability to retrieve up to 51 full lines of text. If you ever forget to turn capture mode On and important data scrolls off the screen, remember MIRROR's retro-capture feature.

A retroactive data capture can be initiated using the `Capture` command. The `Erase` command erases the capture buffer before initiating a retro-capture.

To use the retro-capture feature, type the 'less than sign' (<) as the first argument to the `Capture` command. This tells `MIRROR` that you wish to retroactively capture lines of data. You must follow the less than sign by the number of lines you wish to recapture.
Typing

```
ca < 30 [←]
```

tells `MIRROR` to copy the last thirty lines written to the retro-capture buffer to the main capture buffer.

There does not need to be a space between the less than sign and the number thirty. However, `MIRROR` requires a space between the `ca` and the less than sign. Note that capture mode must be `Off` before you can begin retro-capture.

3.7.4.6 Browsing captured data

The most versatile feature provided by `MIRROR` for browsing captured data is `MIRROR`'s integrated Editor, with which you can browse and edit anything in the capture buffer. You may also bring disk files into the capture buffer for editing purposes. This allows you to browse files you had previously captured directly to disk.

TYPE COMMAND:

The `Type` command provides you with a 'one-way' browse capability. If you wish to browse the capture buffer, use the `Type` command without any arguments

```
ty [←]
```

To browse a disk file, enter the file name as the argument for the `Type` command. A `S` will interrupt the display produced by the `Type` command.

CSTATUS COMMAND:

The `CStatus` command without an argument displays the 'Capture Status' in the Window, telling you the number of lines and characters captured, how much of the capture buffer has been filled, and the amount of remaining free space on the currently logged drive. The `CStatus` command is applicable only to memory captures; you cannot use it unless you are capturing to memory and have the `Capture` status `On` or `Paused` (or `Off` if there is something in the capture buffer).

However, the `CStatus` command is entirely different when used with an argument. With an argument, the `CStatus` command operates as a search command, permitting you to search for any combination of characters up to 67 characters in length. For example, to direct `MIRROR` to search for the phrase 'jumped over the lazy dog' you would type the following:

```
cs jumped over the lazy dog [←]
```

Note:

In this example there is no space between the word 'dog' and the Return, ([←]) since trailing spaces are significant in the search.

PRINTER COMMAND:

The `PRInter` command provides you with a hard copy transcript of everything received by your system. When the `PRInter` is in the `On` setting, a copy of everything displayed on your Terminal Screen will be written to the hard-copy device you have attached to your system. This will continue until you change your `PRInter` setting to `Off`.

PICTURE COMMAND:

The `PIcture` command captures a single screen's worth of data. This command uses the same format as that used with the `CApture` command.

For example, typing

```
pi b:statscr.txt /a [←]
```

appends a picture of your current screen (either the Status Screen or the Terminal Screen) to the file `STATSCR.TXT` on your 'B' drive.

SNAPSHOT COMMAND:

There are two forms of the `SNAPSHOT` command. The first moves data into the snapshot buffer. To do this, you provide `MIRROR` with a single character argument (any character will do). For example, if you wanted to take a snapshot of your current Terminal Screen (for instance, to make a copy of a particularly long command menu), you could type:

```
sn x [←]
```

To display the contents of the snapshot buffer, use the `SNAPSHOT` command without any arguments as follows:

```
sn [←]
```

Unlike the `PIcture` command, you can only take a `SNAPSHOT` of your terminal screen.

3.7.5 Sending Data or Files

The `SEnd` command provides an automated way to transmit an existing text file to another system.

3.7.5.1 Send command

The `SEnd` command should be used to send simple text files to other systems. The `SEnd` command is not designed to send binary files since it lacks sophisticated error detection features. If you need to send binary files, you should use the error detection protocols, which are explained in the next chapter. Data transfer by protocol should always be used when you are communicating with a system that is running `MIRROR`, `CROSSTALK XVI`, or any other system that supports one of the protocols available in `MIRROR`.

To use the `SEnd` command, start a communications session with the other system and prepare the other system to receive data. For instance, if you intend to use the other system's line editor to receive the data, put that editor in data input mode and have it issue its first prompt. Then enter the `SEnd` command (the `Command?` prompt must first appear on the command line; use the `Attention` key [**Esc**] to do this). For example, typing

```
se a:letter.txt [↵]
```

tells `MIRROR` to start sending file `LETTER.TXT` from your 'A' drive to the other system.

3.7.5.2 Character filtering

Just as `MIRROR` allows you to screen out potentially troublesome incoming characters, it also allows you to control what you `SEnd`. Four toggle commands, `Outfiltr`, `TAbex`, `UConly`, and `BLankex`, give you this control. Remember that these switches affect only data transmitted using the `SEnd` command.

3.7.5.3 Flow control

The system with which you are communicating may, from time to time, need to have your system stop sending data so that it can perform some intermediate processing (such as writing its own capture buffer to disk). As mentioned in the last chapter, you can use the `FLOW` command to specify characters that the other system can use to tell your computer to temporarily halt transmission. Usually, `^S` is the 'STOP transmission' character, while `^Q` is used to 'RESTART transmission'. This particular signaling arrangement (a basic protocol, in essence) is called `XON/XOFF` handshaking. If the system with which you are communicating supports `XON/XOFF`, make sure your flow control characters are set accordingly.

3.7.5.4 Communicating with a prompting system

When communicating with a mainframe system, you will often find that the mainframe's text editor is used to receive the data you send. These editors accept input one line at a time, and will insist upon issuing a prompt following every Return sent by your system. Such editors cannot accept the input of an uninterrupted stream of data. To communicate with these systems, `MIRROR` has to act like a human typist.

To mimic a typist, `MIRROR` has to be told how to behave after it has transmitted a line of data. You must specify when or under what conditions `MIRROR` should send the next line of text. The `LWait` (Line Wait) command provides for this function. The `LWait` command has several different formats, including `None`, `Manual`, `Echo`, `Character`, `Prompt`, `Learn`, `Quiet` and `Set`. Refer to the `LWait` help screen for more information.

3.7.5.5 Communications with slower systems

Not all systems with which you wish to communicate use prompts to signal when they are ready for more data. With some of these systems, you may find that not everything you `SEND` reaches its destination. If you notice a loss of characters in transit, you are probably communicating with a system that can't receive data as fast as you can send it (even though your `SPEDS` are compatible). Since you have no way of making the other system faster, `MIRROR` provides you with several ways to slow down your transmission. One is the `LWAIT` command. For very slow systems, there is the `CWAIT` (Character Wait) command.

3.7.6 File transfer using Protocols

3.7.6.1 Introduction

The last two chapters described the transfer of textual data, one line at a time, from one system to another. No mention was made of problems that might arise from the minor errors in transmission caused by noise on the phone lines. Most phone lines are noise free and, with small file sizes, there are likely to be few (if any) transmission errors at moderate baud rates. Also, the consequences of minor errors in the transmission of a text file are usually small.

If you are transmitting numerical data, you may not always be able to distinguish between validly reported data and garbage characters generated by noise on the phone lines. Even if transmission errors were detectable, the recipient may not always know what should have been sent.

When transmitting binary files (the executable code of computer programs), it is impossible to know if garbage characters have been transmitted. The results of even one error in transmission could make the results of the transmission useless.

`MIRROR` solves these problems by permitting you to send and receive both text and binary files using verification protocols. A verification protocol is an operating procedure that provides a standard set of data transmission signals (start transmission, transmission received, transmission rejected, etc.) together with a means of checking the validity of what was sent. Before transmission occurs, of course, both the sending and the receiving system must agree on the protocol to use.

`MIRROR` supports many of the popular protocols in use today: `Crosstalk`, the Hayes verification protocol (`Smartcom`), `KERMIT`, `YMODEM`, and single and multi-file `XMODEM` (with `CRC` or checksum). If the system with which you are communicating supports any of these protocols, you can perform a protocol transfer of text or binary files using `MIRROR`. Since this mode of transfer is faster and more reliable than the simple sending and capturing of data, you should use it whenever possible.

3.7.6.2 Using the Crosstalk protocol

When sending data via the Crosstalk protocol, use the `X M i t` (transmit) command. For example, typing

```
xm testtran.ndc [↵]
```

directs MIRROR to begin sending the file named TESTTRAN.NDC from the currently logged drive on your system to the currently logged drive on the remote system.

Sending data using the Crosstalk protocol is an automated procedure. Simply put your system in Call M O d e and establish a communications link with another system running MIRROR or CROSSTALK. This remote system must be in Answer M O d e.

The `X M i t` command will cause the specified file to be written to the currently logged drive on the remote system, unless you specify an alternate drive to receive the data. For example, suppose you want to transmit a file named REPORT . D O C from your 'B' drive to the 'A' drive on the remote system, you would type

```
xm b:report.doca: [↵]
```

The alternate drive is the final argument on the `X M i t` command.

If the name of the file you send is identical to the name of a file on the other system, the file you send will overwrite and destroy the pre-existing file. You will not be notified in advance, so you should check out the contents of the disk on the remote system before using this command.

While a protocol transmission is in progress, MIRROR keeps you posted on its status with an informative display. As each block (group of characters) of data is sent, MIRROR updates the block and percent complete indicators. Any transmission error will be counted under 'Total errors'. Remember that this is a count of the total number of errors that MIRROR has automatically corrected and not an indication that the other system has received an incorrect transmission.

If you notice a large number of errors, consecutive or otherwise, you may have a bad phone connection. If this should happen, you may wish to cancel the transmission, terminate your telephone connection and re-dial. You may cancel a transmission at any time by pressing the ATtention key. Unless the number of errors significantly increases your telephone or connect time charges, your best choice may be to let the transmission run to completion. After all, the use of the verification protocol ensures that the file will be correctly transmitted in the end, so do not start over unless you have to.

If you experience only errors, you have probably specified the protocol incorrectly. In this case, cancel the transmission and retrace your steps, specifying the proper protocol.

It is also possible to send multiple files, as opposed to a single file, using the `XMit` command. For example, typing

```
xm b:t*.*x [↵]
```

directs `MIRROR` to begin sending all files on your 'B' drive with file names beginning with the letter `t` and having a two-character file type, with any character as the first character and the letter `x` as the second. The receiving system will write all files received to its currently logged drive using the same file names the transmitted files had on the sending system. Again, any pre-existing files on the receiving system that had the same names as the files that were sent will be overwritten.

In the ways shown above, the `XMit` command can be used to transmit files from a `Call Mode` system to another `Crosstalk` compatible system that is in `Answer Mode`. There will be times, though, when a `Calling` system will need to have files transmitted to it. To serve this need, `MIRROR` provides the `Request` command as a counterpart to the `XMit` command.

When the `Request` command is used, all aspects of the data transfer are handled by the `Calling` system just as they are when the `XMit` command is used, with one exception. The difference is that the files are transmitted to the `Calling` system, instead of from. For example, typing

```
rq b:transit.doc [↵]
```

directs `MIRROR` to tell the remote system to transmit, from its 'B' drive, the file named `TRANSIT.DOC`. When this file is received, it will be written to the currently logged drive.

3.7.6.3 XMODEM (single file)

`MIRROR` provides two commands for file transmission using the single file version of the `XMODEM` protocol. These are the `XXmodem` (transmit using `XMODEM`) and the `RXmodem` (receive using `XMODEM`) commands. They work only with the single file version of the `XMODEM` protocol.

You may use the `XXmodem` command to send files using the `XMODEM` protocol. Start a communications session with the remote system and prepare it to receive a file (no description of this process will be attempted here, since there are so many programs that can use single file `XMODEM`). Then enter your `XXmodem` command. For example, typing

```
xx b:report.txt [↵]
```

starts sending the file `REPORT.TXT` from your 'B' drive. Note that there is no way for you to specify a file name or a drive designation to be used on the remote system. The `XMODEM` protocol does not include this information and is, therefore, solely under the control of the remote system.

The `RXmodem` command has the same format as the `XXmodem` command but tells `MIRROR` to begin receiving data into a specific file. For example, typing

```
rx c:archive.doc [↵]
```

directs `MIRROR` to write the data it receives onto file `ARCHIVE.DOC` on your 'C' drive (even if this is not your currently logged disk). If this file already exists, it will be overwritten in the same way as with the Crosstalk protocol.

There are two methods used to verify data transfer using the single file `XMODEM` protocol. These are the `CRC` (Cyclic Redundancy Checking) and checksum calculation methods. With the `RXMODEM` (receive `XMODEM` files) command, `MIRROR` will use `CRC` checking if you select it and if the other system is capable of using the `CRC` method. Otherwise, `MIRROR` will use the checksum method of transmission verification. You may select `CRC` by turning the `CRC` command On. When the `CRC` command is toggled Off (this is the default), the checksum method will be used.

Note:

since it is the receiving system that specifies the method of verification to be used, you need not set a value for the `CRC` parameter when you are sending data. `MIRROR` will only look at this value when you are receiving data.

3.7.6.4 XMODEM (multi-file)

The second version of the `XMODEM` protocol supported by `MIRROR` is the multi-file version of `XMODEM`. The two commands used to provide this capability are `XBatch` (transmit files using `XMODEM`) and `RBatch` (receive files using `XMODEM`). These commands are used to send multiple files at a time from a `MIRROR` system to another system that uses multi-file `XMODEM`. Wildcard characters specify ambiguous file names in the same way as with the `XMit` and `RQuest` commands. For example, typing

```
xb c:* .doc [↵]
```

directs `MIRROR` to transmit all files from your 'C' drive that have a file type of `DOC` to the other system. You cannot supply an alternate drive for the other system to use as you can with the Crosstalk protocol.

The `RBatch` command is the counterpart to `XBatch` and receives files that are sent by a remote system using multi-file `XMODEM`. Since multi-file `XMODEM` transmits file names with the data, it is unnecessary (and erroneous) to specify file names with this command. Thus, all you need to type to receive files to your currently logged drive, using multi-file `XMODEM`, is the following:

```
rb [↵]
```

MIRROR does permit you to specify an alternate drive and/or path to direct incoming files away from the currently logged drive. Thus, typing

```
rb a:\beta\ [←]
```

directs MIRROR to write all files received to the 'A' drive on the BETA subdirectory. Note that the path name must be offset by a pair of backward slashes (\). The XMODEM multi-file protocol uses the same two methods in the same way as the single file XMODEM protocol to verify data transfer. Also note that the multi-file XMODEM protocol verifies not only the data but also the file names transmitted (whereas the Crosstalk protocol verifies only the data).

3.7.6.5 KERMIT

Another protocol gaining increased acceptance for the transfer of files between microcomputers, minicomputers and mainframes is the KERMIT protocol developed at Columbia University. MIRROR provides several commands to manage the transfer of files using the KERMIT protocol. These include commands for communicating with computer systems that are KERMIT servers and with systems that are not (but can still use the KERMIT protocol). In either case if the remote KERMIT host is an IBM VM/CMS Mainframe which supports the KERMIT protocol, then start your session by typing

```
ke ibm on [←]
```

A KERMIT server is a computer system that responds to KERMIT commands issued from remote systems without the need for local operator intervention. Once the communications link has been established, the remote user can issue commands to transfer files from one system to the other completely unassisted.

To transmit files to a KERMIT server, establish your communications link and use the XKermit (transmit files to KERMIT server) command. This command accepts ambiguous file names. For example, typing

```
xk b:data?.* [←]
```

directs MIRROR to use the KERMIT protocol to transmit all files from your 'B' drive with five character file names beginning with the word data. You cannot specify an alternate drive specification for the KERMIT server.

The GKermit command requests the transmission of multiple files from a KERMIT server. For example, you would type

```
gk *.com [←]
```

to request the KERMIT server to transmit all its COM type files to your system. Like the RBatch command, you can specify both an alternate drive and path for incoming data. Thus, typing

```
gk *.* c:\src\ [←]
```

requests that all files from the KERMIT server be transmitted to your system and written to your 'C' drive on the SRC subdirectory.

When you are receiving files from a system that supports the **KERMIT** protocol but is not a **KERMIT** server (such as a **MIRROR** system), you must use the **RKermit** (receive files from **KERMIT** supporting system) command instead of the **GKermit** command. This command has a format similar to that of the **RBatch** command, since file names are transmitted along with the data in a **KERMIT** transmission. Thus, typing

```
rk c:\xyz\ [←]
```

directs **MIRROR** to write any files it receives under the **KERMIT** protocol to your 'C' drive on the **XYZ** subdirectory. Unlike **RBatch**, you may override the transmitted file name by supplying an unambiguous file name (as long as only a single file is sent).

To send files under the **KERMIT** protocol to a system that is not a **KERMIT** server, use the **XKermit** command as explained above. The only difference in using **KERMIT** with a non-server is that you must also prepare the remote system to be ready to receive the file(s) you are going to send.

There are five parameters you may need to adjust when transmitting files using the **KERMIT** protocol. These parameters are:

Binary quoting character,
Control quoting character,
End-of-line character,
Packet size, and
Timeout period.

Unique characters must be used for binary quoting, control quoting, and end-of-line. For a discussion of the usage of each of these parameters, please refer to your **KERMIT** protocol documentation.

MIRROR provides the **KErmit** command to allow you to specify each of the above settings. To change a setting, enter the first character of the parameter as the first argument to the **KErmit** command, followed by a space and then the setting you wish to enter. For example, type

```
ke e @ [←]
```

to change your **KERMIT** end-of-line character to the 'at sign'. This setting may be entered in any of the ways **MIRROR**'s Special Key settings may be entered. To get a List of the current **KERMIT** settings, supply the letter 'l' as the argument to the **KErmit** command as follows:

```
ke l [←]
```

The **KErmit** command can also issue **KERMIT** commands. For instance, to Finish a **KERMIT** server session, enter the letter 'f' with the **KErmit** command as follows:

```
ke f [←]
```

Three other commands can be entered this way: GET, RECV, and SEND. These correspond to the GKermit, the RKermit, and the XKermit commands and have the same function and argument format. For example, typing

```
ke g *.* c:\abc\ [←]
```

directs MIRROR to get all files from the KERMIT server and write them to your 'C' drive on the ABC subdirectory.

3.7.6.6 HAYES protocol

This is the primary protocol supported by the Hayes Smartcom program. The two commands used to transfer files under this protocol are XHayes (transmit files to a Hayes supporting system) and RHayes (receive Hayes files). These control the sending and receiving of files, respectively. For example, typing

```
xh b:april.rpt [←]
```

directs MIRROR to begin sending the file named APRIL. RPT from your 'B' drive to the remote system. Conversely, typing

```
rh c:june.rpt [←]
```

directs MIRROR to write a file sent under the Hayes protocol to your 'C' drive under the name JUNE. RPT.

3.7.6.7 YMODEM protocol

Two commands are available for file transmission using the YMODEM protocol. These are the XYmodem (transmit using YMODEM) and RYmodem (receive using YMODEM) commands. These commands work only if the system on the other end supports YMODEM.

The XYmodem command transmits files using the YMODEM protocol. You can enter either an unambiguous file name or an ambiguous file name designating the file(s) you wish to transmit as this command's argument. For example, typing

```
xy c:*.doc [←]
```

directs MIRROR to transmit all files from your 'C' drive that have a file type of DOC to the other system using the YMODEM protocol.

With XYmodem command, you cannot supply an alternate drive for the other system to use as you can with the Crosstalk protocol. This is not a part of the YMODEM protocol; thus, the remote system will always control the drive that will be used to receive the data. YMODEM does verify the file names that are sent along with the data.

The RYmodem command receives files transmitted by a remote system using the YMODEM protocol. Since this protocol includes file names with the data, you must not specify file names with this command. To receive files to your currently logged drive, using YMODEM, type

```
ry [↵]
```

MIRROR does allow you to specify an alternate drive and/or path to direct incoming files away from the currently logged drive. See the RBatch help screen for more information.

3.7.6.8 Examples

At this point, some examples of data transmission using verification protocols might be helpful.

- a. You want to transmit the file named REPORT.DOC located on your 'B' drive to another system. Your system is running MIRROR, while the other system is running CROSSTALK XVI. The currently logged drive of the other system is drive 'A'. You type the following command:

```
xm b:report.doc [↵]
```

This will transmit a copy of the file REPORT.DOC from your 'B' drive to a file named REPORT.DOC on the 'A' drive of the other system, using the Crosstalk protocol.

- b. On the 'C' drive of a remote MIRROR system, there are several files with a filename of PROJECT (but with different filetypes). You want to request that these files be transmitted to your system using the Crosstalk protocol. In addition, you want these files written to your 'B' drive, even though this is not your currently logged drive. Put your system in Call Mode and the remote system in Answer Mode. Issue the appropriate G0 command to start the communications session. Then type the following command:

```
rq c:project.* b: [↵]
```

- c. You want to transmit all the ASM type files from your 'B' drive (which may or may not be your currently logged drive) to another system. You are running MIRROR. The other system is running neither MIRROR nor CROSSTALK XVI. However, it can receive files using the multi-file version of the XMODEM protocol. Get the other system ready to receive files, then type the following command:

```
xb b:*.asm [↵]
```

- d. You want to receive files sent from another system under the multi-file XMODEM protocol. You want these files written to your 'A' drive on the directory BACKUP. Get the other system ready to send the files, then type the following command:

```
rb a:\backup\ [↵]
```

3.7.7 Advanced Features

MIRROR has many advanced features that are not often needed by the novice and are beyond the scope of this User Instruction Manual Guide. If you want more information on any of the features that are briefly described below, use the `HElp` screen for the command of interest or you may need to purchase the companion Reference Guide which goes into much more depth on all the features of MIRROR.

3.7.7.1 Background use of MIRROR

Suppose you have a fifty page report to transmit. At 1200 baud, this could take half an hour to send, assuming the use of a verification protocol and a good phone line. In the middle of this transmission, you remember that you are due to make a presentation, in fifteen minutes, for which it is vital that you make one last projection using your decision support software. You don't have access to another microcomputer and time is running out. Must you abort your transmission?

With MIRROR, the answer is 'No!' By simply pressing both the left and right Shift keys at the same time, you can run MIRROR in the background. That is, once you have started a MIRROR session, you can put the program into a mode in which it will continue to operate, behind the scenes, at the same time you are using your microcomputer to run some other program (such as a wordprocessor). In addition, you can at any time bring MIRROR back into the foreground to monitor its progress or issue commands by hitting the same two keys. During this time, the other program will be suspended (unless it is another program that can run in the background). This default combination (the left and right Shift keys) can be altered, as explained in the `HElp` file.

A word of caution is in order concerning the background use of MIRROR. If you put MIRROR into the background to perform a task (for example, to transfer files from another computer), and a long distance phone call was required to dial the remote computer, you are incurring long distance phone charges until the computers are disconnected. Therefore, remember to monitor your communications with remote computers, which can be done both manually and by using Script files.

3.7.7.2 Terminal emulation

Terminal emulation allows the microcomputer user running MIRROR to properly display information that is intended for display on various types of computer terminals. It is not uncommon for a mainframe computer program to format its output for a particular terminal type. This is especially true if the program is running on a large mainframe computer.

You can select a terminal emulation by using the `EMulate` command which will display a menu of the available terminals that MIRROR can emulate. MIRROR allows for function key emulation for the emulated terminals. Individual terminal descriptions are found on your MIRROR disk as `' .DOC'` files, for example, `VT100.DOC`.

3.7.7.3 Saving / Reloading / Modifying command files

SAVING PARAMETER SETTINGS:

Once you have the correct parameter settings to communicate with another system, MIRROR lets you **SAVe** them for later use on a disk file. For example, typing:

```
sa tgold [←]
```

writes your current MIRROR settings to the file named **TGOLD.XTK** in command format on your currently logged drive. The drive designation and file type are supplied by MIRROR unless you explicitly specify them.

Note:

The file **STD.XTK** on the disk provided with the PPC, contains a definitive set of default parameters. You may wish to make an extra copy of this file before overwriting it.

RELOADING COMMAND FILES:

Once you have saved a command file, it is a simple matter to **LOad** it and restore its parameter settings. For example, to restore the settings **SAVe**d on the file named **TGOLD.XTK** on your currently logged drive, type:

```
lo tgold [←]
```

MIRROR will provide the **XTK** file type and the drive designation, unless you explicitly include them with the command. When a **LOad** command is issued, MIRROR executes the commands in the command file to restore the specified settings and then executes the **G0** command.

MODIFYING COMMAND FILES:

The easiest way to modify a command file's settings is to **LOad** the file (to restore the command settings), cancel the dialing sequence (by pressing the **AT**tention key), modify the settings using the appropriate MIRROR commands, and then issue a **SAVe** command, directing MIRROR to write the new set of parameter settings to the original file. You can also use MIRROR's integrated text Editor to modify command files, but you may find the previous method more convenient.

Note that many of the example command files are specially set up with a dialogue to establish the correct telephone number to dial. If these files are modified and then **SAVe**d, this dialogue will no longer be invoked.

3.7.7.4 Learn command

The `Learn` command provides a way for `MIRROR` to automatically generate Script files and associated command files. By issuing the `Learn` command, you can create an automatic logon Script file by having `MIRROR` copy the logon sequence of another computer as you manually enter it. `Learn` is a toggle command which you can turn On and Off.

When the learn mode is on, `MIRROR` monitors the keyboard for information and local commands. When you press a key, `MIRROR` uses the key-press as a trigger to record a `Wait String` command for the last ten characters that were received from the remote system. If fewer than ten characters were received, only those characters are used. Concurrently, `MIRROR` is monitoring input from the remote system. Input from the remote system will cause the creation of a `Reply` command with the keystrokes entered by you at the keyboard. This process of constructing `Wait` and `Reply` commands will continue until you turn off the learn mode. Also, `MIRROR` will include in the file any local commands (such as turning `Capture` on) entered by the user through the keyboard.

The `Learn` command should be issued before you begin a session that you wish to save automatically to a Script file. This may be done before the `G0` command is issued or after the carrier is detected, but before you enter any input. `MIRROR` will not record anything until the carrier is detected. Thus, the `G0` command will not be recorded.

You should turn off the `Learn` command after the last user data to be learned has been entered. `MIRROR` saves the automatically generated Script file (file-name.XTS) so that you can use it to exactly reproduce the logon sequence the next time you need to logon to this system. You also need to save the associated command file (filename.XTK); do this by pressing the `Attention` key (usually `[Esc]`) and entering a `Save` command with the file name of the Script file that the `Learn` command just created.

It is often necessary to edit the automatically generated script file. In particular the `Wait String` commands may contain irrelevant details, or even un-repeatable strings such as the time and date!

3.7.7.5 Integrated text editor

An integrated full-screen text editor is an asset to almost any software product, provided it is easy to use and easy to learn. `MIRROR`'s is both, in that it adheres to the standard established by `WORDSTAR` for full-screen editing on microcomputers. This standard relies on a control-key-based command structure that allows all functions to be performed with a minimum of keystrokes, making it easy to use. `MIRROR`'s Editor is easy to learn for `WORDSTAR` users. You invoke the Editor with the `Edit` command.

`MIRROR`'s Editor edits the capture buffer. You are provided with the ability to load disk files into this buffer, but it is still the capture buffer you are editing. All editing changes are made directly to the capture buffer, not (as in `WORDSTAR`) to some intermediate area. Thus, any changes you make will be reflected in the capture buffer, even if you choose to terminate the editing session without saving the information in the buffer to disk. Of course, if it was a disk file that you were editing, such a termination does leave the original file unchanged.

Once in the Editor, you can enter any of the editing commands that follow.

EXIT/SAVE COMMANDS:

- ^kq** This command exits the Editor without saving the data to a disk file (same as pressing the **[Esc]** key). MIRROR will then return you to the screen (Status or Terminal) from which you entered the **E D i t** command. The data in the capture buffer will not be written to disk, but will remain in the capture buffer. However, any changes made will still be reflected in the capture buffer.
- ^kd** This command exits the Editor and saves the contents of the capture buffer to a disk file. MIRROR will then display the name of the file from which your data was loaded into the capture buffer (if any) in the command display at the bottom of your screen. You will be permitted to modify this name, after which you can press Return to save the file. If you leave this disk file name blank, The Editor will treat your command as a **^kq**.
- ^ks** This command saves a copy of the capture buffer to a disk file without exiting the Editor. The Editor will prompt you for a file name in the same way as in the **^kd** command, above. When the disk file has been written (or if you cancel the command), the Editor will return you to edit mode at the same point in the capture buffer.

CURSOR POSITIONING COMMANDS:

- ^s** Move cursor left one character (same as the **[←]**).
- ^d** Move cursor right one character (same as the **[→]**).
- ^e** Move cursor up one line (same as the **[↑]**).
- ^x** Move cursor down one line (same as the **[↓]**).
- ^a** Move cursor left one word (same as the **[←]** key with the **[Ctrl]** key held down).
- ^f** Move cursor right one word (same as the **[→]** key with the **[Ctrl]** key held down).
- ^qd** Move cursor to just after the last character in the line (pressing the **[End]** key also does this).
- ^qs** Move cursor to the first character position in the line (pressing the **[Home]** key also does this).

SCROLLING COMMANDS:

- ^w** Scroll the screen up one line.
- ^z** Scroll the screen down one line.
- ^r** Scroll the screen up twenty-two lines (same as **[PgUp]** key).
- ^c** Scroll the screen down twenty-two lines (same as **[PgDn]** key; note that this is the only key whose function in the Editor differs from its function in editing MIRROR's command line).
- ^qr** Scroll the screen up to the beginning of the capture buffer (pressing **[Ctrl]-[PgUp]** also does this).
- ^qc** Scroll the screen down to the last line in the capture buffer (pressing **[Ctrl]-[PgDn]** also does this).

SPECIAL FUNCTION COMMANDS:

- ^v** Toggle from character insertion mode (the default) to overwrite mode, and vice versa (same as **[Ins]** key).
- ^qg** Toggle graphics display mode on and off. To edit WORDSTAR files, leave graphics mode off (the default setting). In this mode, the high bits of characters will not affect their display, making the display more legible. However, if you want to see graphical characters while editing, use this command to turn graphics mode on. Note that this command does not change high bit settings (as would **^qm**); it only affects the way the Editor displays characters whose high bits are set.
- ^g** Delete the character on which the cursor rests (pressing the **[Del]** key also does this).
- ^t** Delete the remaining characters in this word, starting with the character on which the cursor rests.
- ^y** Delete the line on which the cursor rests.
- ^qy** Starting with the character on which the cursor is positioned, delete the remaining characters on the line.
- ^qm** Strip away the high bits of all characters in the capture buffer. Use this command on files you will be sending to someone who will not be able to use WORDSTAR to process them. Masking the high bits will make a typical WORDSTAR file more legible when displayed outside of WORDSTAR.
- ^n** Insert a line (same as Return in insert mode).
- ^p** This command directs the Editor to place a control character at this character position. The next character pressed after the **^P** becomes the control character (for example, typing **[Ctrl][P][C]**, puts a **^C** in the capture buffer at the character position where the cursor rests). The Editor displays control characters in reverse video.
- ^i** Move to the next tab stop (same as **TAB** key).

STRING LOCATION/MODIFICATION COMMANDS:

- ^qf** Find a character string in the capture buffer.
- ^qa** Find and replace a character string in the capture buffer.
- ^l** Find the next occurrence of the most recently specified character string in the capture buffer.

When you enter either the **^qf** or **^qa** command, the Editor will prompt you to enter the character string you wish located and/or changed in the command display. When entering this string, remember that leading, trailing, and embedded blanks will be significant in making the search. Press Return when you have completely typed the string. If you have entered the **^qa** command, the Editor will then prompt you to enter the replacement string. Either of these strings can be a null character string (press Return without typing any characters; null strings are used to delete strings). You will be prompted to enter any combination (or none) of the following options, by entering the following characters (in any order) and pressing Return. If you wish to have your options listed, enter a question mark and press Return.

Character Option

- b Search back up through the capture buffer, starting at the current cursor position.
- g Search the entire capture buffer, from top to bottom.
- u Ignore case when comparing strings.
- n If you select this option with the `^q a` command, the Editor replaces the first string entered with the second without asking.

If neither option `b` nor option `g` is selected from the above list, the Editor will search down through the capture buffer, starting at the current cursor position and continuing through the end of the capture buffer.

BLOCK MARKING COMMANDS:

- `^k b` Mark the beginning of a block of characters. If the beginning of a block has already been marked, this command will replace it. If both a beginning and ending of a block have been marked and this new mark is below the current end of block mark, the existing end-of-block mark is deleted.
- `^k k` Mark the end of a block of characters. If the end of a block has already been marked, this command will delete the old end-of-block. If a beginning of block already exists and an end-of-block is inserted above it in the file, the Editor reverses their roles (that is, the end-of-block mark is treated as the beginning, and vice versa).
- `^k h` When both the beginning and end of a block have been marked, the Editor will display the entire block of characters in high intensity. This command deletes both block marks and does away with the highlighting. Block marks are also deleted whenever you exit the Editor (block marks are not written to disk files).

BLOCK COMMANDS:

- `^k c` Insert a copy of the block starting at the cursor.
- `^k v` Move the marked block from its current position in the capture buffer to the current cursor position. (Same as the block copy command, except that the original copy of the block is deleted.)
- `^k r` Read the contents of a disk file into the capture buffer starting at the current cursor position. The external disk file is unchanged by this command.
- `^k w` Write a marked block of characters onto an external disk file. The Editor prompts you to name this file.
- `^k u` Upload (transmit to the other system) the marked block. This is the same as a `S E n d` command, except that a `S E n d` transfers a file from disk.

- ^ k p** Print the marked block of characters on the currently assigned printer. MIRROR's `PR i n t e r` parameter must be in one of its two 'off' settings for this command to be executed. Should you wish to cancel a block print command in progress, simply re-enter the command. Note that by typing `p r o f f [↵]` from the command line in MIRROR, you can cancel a block print command that was issued while in the Editor and that continued to execute after you exited from the Editor.
- ^ k y** Delete the marked block of characters.

FUNCTION KEY SETTINGS:

KEY	Function Performed
F1	Interactive Help. The Editor displays its Help Menu..
F2	Start Editing a New File. The Editor will prompt you for a file name in which to save the contents of the capture buffer in the same way that it would had you typed <code>^ k d</code> . After performing this command, the Editor will then prompt you to enter the name of a disk file to be loaded into the capture buffer. The Editor will then load this file and place you in edit mode at the first line of the capture buffer. Pressing the <code>[Esc]</code> key at any point in this process terminates this function.
F3	Exit and Save. (Same as <code>^ k d</code> , if there is no file name.) If there is a file name, however, the Editor will write the capture buffer to the file currently being edited without prompting you for a file name.
F4	Exit Without Saving. (Same as typing <code>^ k q</code> .)
F5	Find String. (Same as typing <code>^ q f</code> .)
F6	Alter String. (Same as typing <code>^ q a</code> .)
F7	Beginning of Block. (Same as typing <code>^ k b</code> .)
F8	End of Block. (Same as typing <code>^ k k</code> .)
F9	Copy Block. (Same as typing <code>^ k c</code> .)
F10	Move Block. (Same as typing <code>^ k v</code> .)

3.7.7.6 Programming with Script

Script is the communications oriented programming language supported by MIRROR. With it, you can set the values of any of MIRROR's parameters just as you could with a command file. It contains conditional statements and you can use these statements to control the logical flow of the Script program when used with Script's branching commands. You can also use the conditional commands to tell MIRROR whether or not to execute a specific command. Script also provides you with the `RE p l y` command, which is used to send commands and responses to a remote system. Anyone who has had experience with a computer programming language should be able to make productive use of Script in a short time. However, it is a programming language and its use will require programming skills.

Using the Script commands, you can write moderately sophisticated communications programs. You can also use Script to write menu-driven procedures to help less knowledgeable users dial into various computer systems. An example of menu-oriented Script file is the FILETRAN command file provided with MIRROR. You may find it instructive to browse this file with MIRROR's Editor, to get a feel for Script programming technique.

Script program files can be created using MIRROR's Editor. When choosing a name for your file, remember that it should have a file type of X T S (for example, MYSCRIPT.XTS).

You can include any MIRROR command in a Script file (the GO and RD i a l commands have a special effect that is discussed in each command's HE l p screen). Use the same format for a Script file command that you would use when entering the command on the Status Screen (the F I l t e r command is a special case and is discussed in its HE l p file). If you don't include enough parameter information with the command, MIRROR will prompt you for it in the usual way during the execution of the Script file. A transaction log representing the execution of a Script file can be kept using the T L command.

When MIRROR is told to execute a Script file using the D O command, MIRROR will perform the commands in a Script file one after the other in the order that they appear in the file. Should MIRROR encounter a branching command (such as a J U m p), it will perform as directed. Thus, you have the ability to make MIRROR skip over whole sections of code, or even cause MIRROR to retrace some or all the steps of the program.

MIRROR permits you to include comments within Script files for documentation purposes. MIRROR will interpret any line as a comment if the line begins with a semicolon. Along with the colon (used to separate two commands on the same line) and semicolon, MIRROR gives special treatment to two additional characters when included in a Script file. When you need to transmit a Return to another system in the RE p l y command, use the vertical bar (|) character. If you need to send a controlled character for execution on the other system, precede the letter with a caret (^).

SCRIPT COMMAND SUMMARY:

While all MIRROR commands can be included in a Script file, a few are specially designed for this purpose. The following is a short reference list of the commands specially designed for Script file processing:

A B o r t	Cancel the execution of the SCRIPT file.
A L a r m	Select and/or sound an alarm tone.
A S k	Prompt for input.
C L e a r	Clear the Screen or Window.
D O	Start or resume execution of a SCRIPT file.
E N D	Same as the A B o r t command above.
I F	Conditional branching.
J U m p	Branch to a L A b e l e d line.
L A b e l	Assign a label to a line.
M E s s a g e	Display a message to the local system's screen.

REply	Send a string of text to another system.
RTurn	Return control to the previous (calling) Script.
RWind	Restart at the beginning of this Script file.
SBreak	Send a BREAK signal to the modem.
Screen	Switch from Terminal to Status Screen and vice versa.
SKip	This command is identical to the JUmP command.
WAit	Wait until the occurrence of a given event.
WHen	Execute a specific command whenever a given character string is received.

SAMPLE SCRIPT PROGRAM:

Script files are used to automate manual procedures. A useful way to go about programming a Script file, then, is to get a transcript of the manual procedure you intend to automate. To obtain such a transcript, use MIRROR on a system with an attached line printer and with MIRROR's PRinter setting in the On position or use MIRROR's Le a R n command. In this mode, go through the manual procedure that will be automated. When you have done this, you will have a transcript from you can save the command file and you will have working command and Script files which will reproduce the manual procedure.

The sample program below involves signing on to a mainframe system to get a copy of the terminal output of a program that will be executed on that system. When all output has been received, an alarm tone will be sounded and Script processing will end. It is assumed that all necessary communications parameters have been set before this Script file is started.

```

;
; When the prompt 'logid:' is received from the remote system,
; respond with the string 'acctgp' and a Return.
;
wa s 'logid:'
re acctgp
;
; When the prompt 'password:' is received, respond with the
; string '31416' and a Return.
;
wa s 'password:'
re 31416
;
; When the prompt 'ready' is received, turn capture mode on.
;
wa s 'ready'
ca on
;
; Provide that a Return be sent each time the string
; '***' is received from the remote system, and send the
; command to start the mainframe program running. Then
; wait until the first character has been received.
;
wh '***' re |
re reptpgl
wa prompt 1

```

```

;
; When the line goes quiet for 15 seconds, sound alarm tone
; number three and terminate Script processing.
;
wa q 150
al 3n
ab

```

Note the use of comment lines (those beginning with a semicolon) in this Script file and how they improve readability.

3.7.7.7 Command Completion status

Often when using a Script file with MIRROR, it would be convenient if you could test for the successful completion of a particular command within the Script file that is being executed. For instance, if a particular command was not successful, the Script file could be written to retry the unsuccessful command. To allow this, MIRROR has a Command Completion Register which is indicated by the number sign (#). MIRROR updates this register with a completion code following all file transfer commands and most commands where disk files are opened (CAPture, WRite, etc.).

The IF and JUmP commands use to check the command completion register. The JUmP command uses the register by including the number sign (#) as a part of the object label (similar to the use of the '@' when JUmP is used together with the ASK command).

3.7.7.8 Transaction Log

MIRROR allows the user to record a log of transactions performed by MIRROR whether entered from the local keyboard, by the remote user or via a Script. If the transaction log feature is enabled, all commands will be written to a file along with the date, time and source of the command. You can toggle the TL command On and Off as needed.

3.7.7.9 Answer mode

To act as a host computer system, use the MOde command to put your system into Answer MOde as follows:

```
mo a [←]
```

As with all host systems, you must make preparations before you open shop as an Answer MOde system (that is, before you issue the GO command). These preparations are not complex and consist of nothing more than setting the appropriate communications parameters that you specify when acting as a Call MOde system. This task is even easier in Answer MOde since you call the shots (although you will want to select settings which will be easy for your users to match, in terms of SPeed, PARity, etc.). You will also wish to use the NAME command to specify a site identification message that will be displayed to each remote user when communications are established. Remember that the remote user dialing into a MIRROR Answer system should have his DUPlex set to FULL. After having set the above parameters, you will be ready to enter your GO command. Your Answer MOde system will then be ready to receive calls.

3.7.7.10 Password maintenance utility

PASSWORD, MIRROR's Password Maintenance Utility comes with the package. This is a separate program that you can run to establish a list of users who are authorized to use your MIRROR system, together with their personal passwords and authorization levels. You can also provide (for each user) the name of a Script file that is to be executed at the start of each communications session. The set of password information for a given user is called the user's security profile.

MIRROR's Password Maintenance Utility includes a form of the `ACccept` command. In this context, however, MIRROR provides an `ACccept` setting for each user who is authorized to use your system. When extended password protection has been selected, MIRROR's basic `ACccept` setting is overridden by the settings in force for the individually authorized users.

To signal MIRROR that you have chosen to use its extended password protection features, enter a password file name as your `PWOrd` parameter. A password file is a set of user security profiles you will create using `PASSWORD`. Any file name can be used, although MIRROR will supply file type `PSW` if no file type is supplied. Users will be given three tries at logging in before the call is rejected as an attempted security violation and the connection terminated.

3.7.7.11 Function Key usage

MIRROR permits you to use your function keys in place of character strings. In addition to the ten 'normal' function keys, MIRROR permits you to access thirty additional function keys by using the `[Alt]`, `[Ctrl]`, and `[Shift]` keys. For instance, holding down the `[Alt]` key, while simultaneously pressing function key one directs MIRROR to play back the character string assigned to alternate function key one. The `[Ctrl]` and `[Shift]` keys work the same way. Function key settings are among the parameters that can be `SAve d` to a file, and then `LOAd e d` back again. You can therefore develop a specific set of function key settings to match each remote system with which you regularly communicate. Each function key can be given a string of up to forty characters.

FUNCTION KEY DEFINITIONS:

The Function Key (`FKeys`) command displays and changes the character strings assigned to MIRROR's function keys. To assign a character string to a function key, supply the number of the function key as the first argument to the `FKeys` command, and follow it with the character string you want assigned to the function key. For example, to assign the character string `Ralph` to `[Alt][F1]`, you could type:

```
fk a1 Ralph [←]
```

FUNCTION KEY EDITING:

MIRROR gives you the ability to edit any character string in the command line. This can be very useful in the editing of function key character strings, which tend to be lengthy and which may contain exotic characters. To bring the contents of [Alt] [F1] into the command line for editing, you would type

```
fk a1 [←]
```

Once the character string is in the command line, you can use MIRROR's line editing features to make any necessary changes.

3.7.7.12 Remote command capability

Remote command capability may be used by other users who have dialed into a MIRROR Answer MOde system that is on-line. The operator of this Answer system can allow remote users to request directories, check space available, and/or send and receive files under various protocols. Almost all operations can be achieved from the calling system. Therefore, the Answer system can be left totally unattended. Before you can enter a Remote Command to the remote system, you must first type the COmmand key. Note that the COmmand key you must type to signal the remote system is not the key specified by the COmmand parameter on your system. Instead, it is the key specified by the COmmand parameter by the person who brought up the remote system.

3.7.7.13 Operating System-like commands

The following MIRROR commands act similarly to the MS-DOS operating system 'direct' commands (for example, DIR, TYPE):

CDir pn	change to new sub-directory pathname
Dir x:afn /y	list directory of files
Drive x:	select new drive as default
ERase x:afn /y	erase file(s) or capture buffer
PRinter p	echo terminal display to printer
Type x:ufn	list text file on screen

3.7.7.14 Screen command

If you have a color display system, you can make use of MIRROR's color control through the SCreen command. With this command, you can adjust the colors MIRROR will use to display commands, settings, and terminal output. You can control the colors of the characters MIRROR highlights, the colors of MIRROR's normal intensity characters, and the colors used on the command line. The SCreen command also permits you to select the desired screen type from within a Script file. You may use this command with the letter 't' as an argument to select the Terminal Screen as your current display screen, or the letter 's' to select the Status Screen, or the letter 'a' to select the alternate screen.

3.7.7.15 Setting the special keys

MIRROR allows you to select four keys on your keyboard to perform four special functions. These keys cannot be set to any key whose character is displayed when pressed, such as an alphabetic or numeric character. These keys are the ATtention key, the SWi t c h key, the BR e a k key, and the remote COmmand key. MIRROR displays the current key settings on the Status Screen. The first three keys perform their special functions when pressed by the local user. The remote COmmand key, on the other hand, performs its special function when pressed by a user on a remote system communicating with an Answer MOde MIRROR system.

3.8 Alphabetical Command Summary

For more information on any of the following commands, use the `HElp` command followed by the first two letters of the command for which you need more information.

`ABort`

Cancels the execution of the currently active Script file.

`ACcept argument`

Limits the effect that remote commands can have on disk files.

`ALarm argument(s)`

Selects and/or sounds one of the alarm tones (numbered 0 through 7).

`ANswback On or Off`

If the `ANswback` command is `On` (which is the default), `MIRROR` responds to a `^E` from the remote system by sending the contents of the local system's function key 4 (F4). If it is `Off`, the `^E` will be ignored.

`APrefix string`

Specifies a string of characters which `MIRROR` issues to the modem when the `GO` command is entered while in `Answer Mode`.

`ASk message`

Prompts you for input to be used in Script file processing.

`ATtention key`

Changes the key used to bring up the `Command?` prompt when you are on the `Terminal Screen`.

`BAckgrnd argument`

Alters the key combination used to enter or return from background use. Used in a Script file to begin `MIRROR`'s background (foreground) capability.

`BKsize number`

Specifies 'block' size when using the `Crosstalk` protocol.

`BLankex On or Off`

When this toggle is `On`, `MIRROR` will replace any blank line in the text you are `SEnding` with a line consisting of a single space.

BReak key

Changes the key used to send a break.

BYe

Terminates the call in progress, but stays in **MIRROR**.

CApture file name or argument

Stores received information either on a disk file or temporarily in memory.

CDir argument(s)

Allows you to change the subdirectory for the currently logged drive.

CLear

Erases the screen.

COmmand key

Changes the key used by the remote user to get your system's attention to enter commands.

CProgress parameters

Provides you with a way to monitor the dialing process, if your modem supports this capability.

CRc On or Off

Selects the CRC algorithm for data verification with the **XMODEM** protocols (single and multi-file).

CStatus string

Provides you with a way to monitor the capture buffer.

CWait argument

Deals with slow systems during the **SEnd** process.

DAta number

Allows you to select the number of data bits in each character.

DEbug argument

A character filtering command which is useful in debugging incoming transmissions.

Dir x:afn /y

Lists alphabetically the names of all the files on drive 'x' that satisfy the ambiguous file name (afn).

DNames number

Specifies the number of directory entries for sorting purposes.

D0 file name

MIRROR will begin executing the Script file named.

DPrefix string

Allows the user to specify a string of up to twenty characters which are to precede the number in the dial string.

DRive x:

Makes disk drive 'x' the new default (or current) drive.

DSuffix string

Allows you to specify a string of up to twenty characters which are to be sent after the phone number.

DUPlex Full or Half

Allows you to select the duplex mode of the transmission.

EDit

Uses either MIRROR's integrated text Editor or the editor named in the **EPa t h** command.

EMulate argument

Allows you to properly display information that is intended to be displayed on various types of computer terminals emulated by MIRROR.

ENd

Cancels the execution of the currently active Script file and is identical to the **ABo r t** command.

EPa t h parameter(s)

Determines the Edit Path MIRROR will follow when you enter the **ED i t** command.

ERase x:afn /y

Erases all files on drive 'x' that satisfy the ambiguous file name (afn). Also used to erase the capture buffer.

Filter parameters

Tells MIRROR which control characters to discard when the `INfilter` switch is in the `On` setting.

FKeys string

Assigns character strings to the forty function keys supported by MIRROR, and to display the character strings currently assigned to these keys.

Flow characters

Specifies characters to `STOP` and `RESTART` transmission.

GKermit afn

Requests the ambiguous file name (afn) from a KERMITS server.

GO argument(s)

Initiates communications with another system.

HAndshak On or Off

If your modem requires the use of the request-to-send (RTS) and clear-to-send (CTS) signals from your RS-232 interface, you must toggle the `HAndshak` switch to its `On` setting. Most modems do not need these signals, so MIRROR's default setting for this switch is `Off`, which is correct for the PPC.

HElp command name

Displays a series of screens with all available commands. `HElp` followed by the first two letters of another command provides helpful information on that particular command. The `.HLP` files must be either on the currently logged drive or on a drive and/or directory specified in the `SYspath` command.

IF argument

Conditionally executes Script file commands based on user input or on the status of your modem's carrier signal.

INfilter On or Off

Tells MIRROR to discard certain control characters (those set to the `No` setting in the Filter table) when the `INfilter` switch is in the `On` setting.

JUmp location

Controls the flow of a Script program.

KErmit argument

Allows you to specify KERMITS protocol settings and issue KERMITS commands.

LAbel name

Assigns a name to a line of a Script file.

LBreak n

Sets the length of the transmitted Break signal to 'n' milliseconds (.001 seconds).

LFauto On or Off

When On, directs MIRROR to insert a line feed after each Return it receives from the remote system.

LIs t

Displays certain MIRROR parameter settings.

L0ad filename

Loads and executes the command file from the currently logged drive.

LeaRn On or Off or filename

Provides for automatically generating Script and associated command files for logging onto remote systems.

LWait argument(s)

Controls the SEnd command's transmission of lines of text.

MEssage

Displays a line or lines of text to the screen.

MInit string

Permits the user to initialize the modem with a string of up to twenty characters.

M0de argument

Allows you to select the 'role' (Call or Answer) that MIRROR will play in a connection.

NAmE sadr

Enters the site address (sadr) that you are calling.

NUmber nstr

Permits you to specify the phone number (nstr) of the remote site that MIRROR will attempt to dial when you enter the G0 command.

OUtfiltr On or Off

Permits you to direct **MIRROR** not to **SEnd** any line-feeds or nulls which the file you are **SEnding** may contain.

PArity argument

Allows you to select the parity of characters.

PIcture filename

Captures a single screen's worth of data.

PMode n

Identifies the type of system with which you are communicating.

POrt n options

Allows **MIRROR** to use the appropriate port (n) for communications. The **PPC** internal modem is port 2.

PRinter argument

Controls the echoing of displayed information to the printer or a file to print.

PWord filename

Invokes **MIRROR**'s password protection features.

QUit

Terminates the call in progress and returns to the operating system.

RBatch x:\pathname\ /option

Receives files transmitted by a remote system using the multi-file **XMODEM** protocol.

RDial number

Allows you to specify the number of times an auto-dial modem will automatically redial the **NUmber** if the connection is not completed in the first dialing.

REply remote command

Using **REply**, you can send commands to, and respond to prompts from, a remote system.

RHayes x:ufn

Receives files transmitted from a remote system using the **Hayes** protocol.

RKermit x:\pathname\

When you are receiving files using the **KERMIT** protocol from a system that is not a **KERMIT** server (such as a **MIRROR** system), you must use the **RKermit** command instead of the **GKermit** command.

RQuest x:afn y:

Issued from a **Call M0de** system to **RQuest** the transmission of a file or files. Ambiguous file names (afn) can be specified from an **Answer M0de MIRROR** or **CROSSTALK XVI** system using the **Crosstalk** protocol.

RTurn

Ends the current **Script** and returns control to the previous (calling) **Script**.

RUn program name

Executes another program (pn) from within **MIRROR**.

RWind

Directs **MIRROR** to restart the currently executing **Script** file from its beginning.

RXmodem x:ufn /option

Receives a file transmitted using the single file version of the **XMODEM** protocol.

RYmodem x:\pathname\ /option

Receives files transmitted by a remote system using the **YMODEM** protocol.

SAve x:ufn

Saves a command file, representing all **MIRROR**'s parameter settings.

SBreak

Used from within **Script** files to send a **BREAK** signal to your modem.

Sscreen argument(s)

Allows you to select the colors that will be used when **MIRROR** is run on a color monitor. Selects whether terminal, status or alternate screen is displayed.

SEnd filename

Sends simple text files to other systems.

SKip location

The **SKip** and **JUmp** commands are identical, and can be used interchangeably.

SSnapshot argument

Makes a temporary copy of your Terminal Screen.

Speed number

Allows you to select a new baud rate.

STop number

Allows you to select the number of stop bits on each character transmitted.

Switch key

Changes the key used to switch between the Status Screen and the Terminal Screen.

SYspath x:\pathname

Allows you to specify the drive 'x' and subdirectory (pathname) where the MIRROR help file, emulation files, command files and/or script files will be if they are not on the currently logged drive.

TAbex On or Off

When sending a file with tab characters, you can toggle **T A b e x On** to expand them into the proper number of spaces.

Timer On or Off

Toggles the on-line **T I m e r** switch On and Off.

Turnarnd key

Changes the key used to indicate the end-of-line character from its default setting (the Return key).

TYpe x:ufn

Lists the contents of the capture buffer or a disk file (ufn) from drive 'x' to your screen.

TLog Off or filename /argument

Allows you to capture the transactions of the session.

UOnly On or Off

When this toggle is set On, MIRROR will convert all lower case characters to upper case before sending them to another system.

Version

Displays the MIRROR version number and release date on the Command Line.

VIdeo argument /options

Allows the user to control the amount of snow and flashing on the video display.

WAit argument(s)

Causes Script processing to pause and wait until a specific event occurs.

WHeN argument(s)

The WHeN command, like the IF command, causes the execution of the rest of the commands on its line when its condition is met.

WRite filename

Tells MIRROR to write the data in the capture buffer to a disk file.

XBatch x:afn /option

Transmits files using the multi-file version of the XMODEM protocol.

XDos

Allows the user to leave MIRROR without disconnecting the call in progress.

XHayes x:ufn

Transfers files under the Hayes verification protocol.

XKermit x:afn

Transmits files to a KERMIT-supporting system.

XMit x:afny:

Transmits data via the Crosstalk protocol.

XXmodem x:ufn /option

Sends files using the single file version of the XMODEM protocol.

XYmodem x:afn /option

Transmits files using the YMODEM protocol.

APPENDICES

Appendix A:

BUYING AND INSTALLING PROGRAMS

The AMSTRAD PPC is supplied to you with 'system' software - the 'tools' needed to use your computer - but not with 'application' software - the programs you will need to carry out computational tasks. You will need to buy from your dealer either a suitable programming language COMPILER, such as BASIC, with which to write your own programs, or application programs ready for use.

This appendix provides a guide to selecting the programs to run on your PPC and to avoiding potential pitfalls, by covering:

- * **What makes a program suitable to run on your AMSTRAD PPC**
- * **Which operating system to run the program under**
- * **How to go about installing a program on your machine**
- * **Making programs convenient to use**

A.1 Suitable programs

The AMSTRAD PPC's compatibility with many other 16-bit micros means you can choose from a wide range of programs bought 'off the shelf' from your computer dealer. Its compatibility with the IBM PC and PC compatibles means you can use programs written for machines running the PC-DOS operating system. These could be computer games, accountants' spreadsheets, word processing programs or project planners.

Many of these programs will run perfectly on the AMSTRAD PPC. Other programs will run but won't always produce the results you expect. They may appear not to control the screen properly or they may not respond correctly to the keyboard. Such programs will need to be 'installed' or 'configured' for your PPC display type and/or the peripherals (printers etc.) you have as part of your PPC system. How to go about installing a program is described in section A.3.

Programs for the AMSTRAD PPC should be

- * **Written to run under the PC-DOS or MS-DOS operating system**
And
- * **Supplied on 3 $\frac{1}{2}$ inch 720 KB floppy disks**

Choose for preference ones which have been set up either specifically for the AMSTRAD PPC, or for the IBM PC with a colour screen. This ensures that the program will control the screen properly and interpret what you type correctly.

Your dealer will be able to tell you whether a suitable version of the software you want to use exists or if it will need to be installed. Installing a program is a straightforward task, particularly if it is supplied with a special installation program, but it may require detailed background knowledge.

A large amount of software won't need to be installed on your PPC.

Avoid, unless there is no other option, programs that have been written specifically for a computer other than the IBM PC or the AMSTRAD PC or PPC. Such software will probably need to be modified before it can be run on your PPC and this may not be possible without specialised help. Check with your dealer that this is not the case.

Note:

You will often find BASIC programs on sale alongside DOS. These can only be used on your AMSTRAD PPC if you have loaded both the BASIC language which the program was written for and the operating system which the BASIC language was designed to be used with.

The BASIC program may require either a particular BASIC or a Microsoft-compatible BASIC, in which case you will also need to acquire a copy of that BASIC in a version that runs under either MS-DOS or PC-DOS. For example, you can run BASIC programs originally developed for the AMSTRAD PCW machines by running the DOS version of Mallard BASIC on your AMSTRAD PPC. (Programs developed using AMSTRAD BASIC on the CPC machines will, however, need to be converted.)

Note:

IBM Personal Computer BASIC won't run. It only runs on IBM PCs.

A.2 Selecting which operating system to use

Alongside choosing an application program, you need to select which system software to use when running the program. Some versions of programs are supplied on disk together with appropriate operating system software: these are known as Turnkey versions of the programs, because they let you just turn your machine on and use the programs. (We describe how to make Turnkey versions of the programs you buy in section A.5, below.)

The remainder require you to use your AMSTRAD PPC SYSTEM disk to load the necessary system software into your machine before you run the program.

A.3 First steps

a. Copy the disks you have just bought

The disks you have just bought are both valuable and vulnerable to accidents. You should therefore always work with copies of disks rather than with the original disks, if this is at all possible.

So the very first thing to do is to make copies of these disks. The instructions are given in section 1.6.1. Then store away the original disks as your Master copies for use only to make new copies from in case of accidents.

Note:

Some programs are copy protected - you can't copy these by the standard method. However, you may be able to copy all but a part - how to do this will be described in the program's own user guide.

b. Install the programs, if this is necessary.

A program may need installing if:

- * **your version of the program isn't specifically marked as being for the AMSTRAD PPC and compatible with the type of display. Refer to appendix D if you are in any doubt about the compatibility of your display.**
- * **you have added to your AMSTRAD PPC system in any way, for example, by adding a printer**

Your dealer should be able to tell you whether you will need to install any program you buy, though you can also discover whether a program needs to be installed by running it. If, for example, it appears not to control the screen properly or text sent to the printer doesn't come out right, then you need to install the program.

If an installation program is provided with the software, you may well be able to install the program yourself. Software suppliers typically include a fairly detailed set of instructions with any software package that needs installing and we strongly advise you to read this.

Do be careful to check whether the installation instructions were written for the AMSTRAD PPC or for some other computer, for instance the IBM PC. If it was written for the IBM PC, then information about the keyboard and the screen will be correct but instructions about where to find particular external commands on the SYSTEM disks will probably be wrong. You may also find you have an .EXE file where the instructions refer to a .COM file or vice versa: in this case, use the .EXE file as if it were the .COM file referred to.

If there is no installation program, you will probably need specialised help.

The job of an installation program is to set some parameters within the main program according to information you supply about how the AMSTRAD PPC works. The information about the AMSTRAD PPC you might need when you run an installation program is given in appendix D.

However, the form in which this information will be requested varies from application program to application program. Sometimes you will be asked to select from a range of standard options (which options to choose are detailed in appendix D), but at other times to press particular keys or to type in the codes given in appendix D.

If you run into any problems, consult your dealer.

A.4 Creating disks geared towards a particular program

The way to make a program most straightforward to use is to put the copy of the program you use daily on a disk with copies of support programs it calls on (for example, MS-DOS external commands) but nothing else. Any space left over on the disk can be used for the program's data files, if it doesn't create or use too many of these, but otherwise it is best left blank with separate disks used to store the program's data.

Mixing up different programs on the same disk may reduce the number of disks you need to buy but it also causes a lot of hassle! You really are much better off with a separate disk for each application.

The problem then with a new program is to what to put alongside the main program on the disk.

Your best course of action is to spend some time studying the program's user guide and discovering from this which programs and external commands the main program needs to have access to. The user guide will often tell you precisely which files you need to have on the program disk.

If you have room on the disk or if you are able to divide the program files among a number of disks, you can also add generally useful external commands such as the disk copier program. However, in practice, you are likely to have to work out which files are essential and which are something of a luxury.

What each program needs is specific to that program so it is not possible to give general advice. But to help, appendix B is a table listing all the files supplied on your AMSTRAD PPC software disks, explaining what each file is used for.

Another possible source of advice is, of course, the computer dealer you bought the program from or a friend or colleague who is already using the program. But don't be afraid to try out a combination of files that seems to be appropriate from what you have read. It is unlikely to take you more than a few goes to get the program working and there is a fair chance you will get it right first time!

A.5 Making a DOS program convenient to use

A DOS program becomes particularly convenient to use if you put into a special Batch file all the instructions needed:

- **to set up the search path(s) that make the program files and data files appear as if in the default directory (see section 2.4.2.4)**
- **to set up your PPC's input and output devices in the right way for the program (see section 2.7.1)**
- **to run the program itself**
- **to carry out any supplementary actions, such as making back-up copies of files that the program has created and deleting any files that you don't want to keep**

(Batch files are described in section 2.4.4.)

You can set up such a Batch file either from scratch or by editing an existing Batch file. In either case, the AMSTRAD PPC's text editor RPED is probably the best tool to use: RPED is particularly suited to creating and changing short files such as Batch file. (How to use RPED is described in sections 1.6.8)

By the way, always be careful not to give your Batch file the same filename as any other program file in the same directory (the filename is the part of the file's name before the full stop), because MS-DOS will always run the program in preference to the Batch file. So if, for example, you set up a Batch file to run a program stored in a file called SPREAD.COM, don't call your Batch file SPREAD.BAT. Call it something like MYSREAD.BAT instead.

Once you have set up such a Batch file, the only command line you need to type in order to do all this work is one made up of the filename of the Batch file and any parameters the Batch file requires.

A special way of making a program convenient to use is to make it run automatically immediately after the operating system software has loaded. Such a program is known as a Turnkey program - from the idea of just turning on and running. It is particularly useful if you want other people, who don't necessarily understand how to run a program on your PPC, to use the software you have bought. You need only to tell them to turn the PPC on, to follow the instructions on the screen (in particular, to insert the disk you have given them when they see the message 'Insert a SYSTEM disk') and to follow the instructions in the program's own user guide.

In order to make a program operate in this way you have to do the following things:

a. Prepare a new blank disk and store the operating system software on it.

Under MS-DOS, this means, to start with, formatting the disk using the /S option of the FORMAT command (see section 2.6.1). It also means copying the ASSIGN.SYS and CONFIG.SYS files from your PPC SYSTEM disk to the new disk, plus the RAMDRIVE.SYS, ANSISYS and KEYB.EXE files if your current CONFIG.SYS or AUTOEXEC.BAT call on these files.

b. Copy the files from your program disk onto this disk.

c. Edit your Batch file to run the program so that it includes the commands from your current AUTOEXEC.BAT file that you will need while running the program.

If you don't use a Batch file to run your program, copy the AUTOEXEC.BAT file from your Startup disk to the Root directory on your new disk. Then edit it so that the last line is the command line you use to run your program.

d. Change the name of the Batch file to AUTOEXEC.BAT and store it the Root directory of your program disk.

You can then use this disk in the Startup procedure as a Startup disk. Test that the software now works as Turnkey Software by inserting the disk in Drive A of your PPC and holding down the [Ctrl] and [Alt] keys while pressing the [Del] key. If all is well, the welcome message of your purchased software will now be displayed on the screen at the end of the Startup procedure.

Postscript

Your software is now ready for use and instructions on how to run it are given in sections 1.5.2 and 1.5.3.

Appendix B:

PPC SOFTWARE DISKS

The SYSTEM disk is supplied with all PPCs. PPC640 is supplied with a second disk containing the MIRROR II communications software.

B.1 Disk 1: THE AMSTRAD PPC SYSTEM DISK

This is a complete listing of the files stored on the SYSTEM disk supplied with your AMSTRAD PPC and explains the use each file has - to help you know which files to have available on the disk to use for any particular job.

The MS-DOS operating system software (hidden files)

IO.SYS
MSDOS.SYS

The MS-DOS internal commands

COMMAND.COM

System configuration files (see appendix E, section E.3)

ANSI.SYS
CONFIG.SYS
COUNTRY.SYS
DATETEST.COM
DRIVER.SYS
GRAFTBLE.COM
GRAPHICS.COM
KEYB.COM
KEYBOARD.SYS
LCD.COM
MODE.COM
PRINTER.SYS
RAMDRIVE.SYS

The Startup procedure

AUTOEXEC.BAT

The MS-DOS external commands. See sections 2.4 to 2.8.

APPEND.COM
ASSIGN.COM
ATTRIB.EXE
CHKDSK.COM
COMP.COM
DEBUG.COM
DISKCOMP.COM
DISKCOPY.COM
EDLIN.COM
FASTOPEN.EXE
FC.EXE
FIND.EXE
FORMAT.COM
JOIN.EXE
LABEL.COM
MORE.COM
NLSFUNC.EXE
PRINT.COM
REPLACE.EXE
SELECT.COM
SORT.EXE
SUBST.EXE
SYS.COM
TREE.COM
XCOPY.EXE
4201.CPI
5202.CPI

B.2 Disk 2: Softklone's MIRROR II

Data Files

\SEND.TST

\MAILBOX\FROMHOME.TST

The Mirror II Software

MIRROR.EXE
PASSWORD.EXE
MIRROR.CFG
VIEWDATA.MAF

**Special (compressed) file of Mirror II help messages. Cannot be
TYPED or PRINTed**

MIRROR.HLP

Files used for demonstration of MIRROR

STD.XTK

NEW USER

BRKTHRU.XTS
CSERV.XTS
DELPHI.XTS
DOWJONES.XTS
LEXIS.XTS
MCIMAIL.XTS
NEWSNET.XTS
OAG.XTS
SOURCE.XTS
VUTEXT.XTS
NEWUSER.XTK
VID_MONO.XTK

File Transfer (modem)

ANSMODEM.XTK
CALLHOME.XTK
CALLHOME.XTS
CHPREFIX.XTS
FILETRAN.XTS

File Transfer (direct connect)

ANS232.XTK
FILETRAN.XTK

Documentation for Terminal Emulation. Can be TYPed or PRINTed

ADDSA1.DOC
ADM3A.DOC
ANSI.DOC
D210.DOC
IBM3101.DOC
PRESTEL.DOC
TI940.DOC
TV912920.DOC
TV925.DOC
VIP7200.DOC
VT100.DOC
VT52.DOC

Terminal Emulation Files

ADDSA1.MEF
ADM3A.MEF
ANSI.MEF
D210.MEF
IBM3101.MEF
TI940.MEF
TV912920.MEF
TV925.MEF
VIP7200.MEF
VT100.MEF
VT52.MEF

Appendix C:

THE AMSTRAD PPC CHARACTER SET

Overleaf is a table giving all the characters available, together with the keys you need to press to put each character onto the screen or into a file. There are a total of 254 different characters.

The characters are listed in the order of their internal code, that is the 8-bit number that the PPC interprets as the character you want to display. (The character set is based on 8-bit internal codes: that is how 254 different characters can be displayed.)

An internal code is always written either as a Decimal number or as a Hexadecimal number: the table gives both. Hexadecimal numbers are numbers to the base 16. When a character code is represented as a Hexadecimal number, it is normally written as &H, followed by a two-character number. The two-character number is actual hexadecimal number. These characters can be one of the digits 0...9 or one of the letters A...F.

Sections C.1 and C.2 below explain how to discover from this table both the key or keys to press to get the character you want and the internal code of any character.

Not all programs have been written for an 8-bit character set: many have been written for the older 7-bit character set that only gave a range of 127 different characters. You can always tell if you are using the 'wrong' character set because every so often what the program has written on your screen will appear to be rubbish.

C.1 To find the key that produces a particular character

	DEC	0	16	32	48	64	80	96	112
	HEX	0	1	2	3	4	5	6	7
0	0	BLANK	▶ 16	BLANK SPACE	∅	Ⓐ	P	·	p
1	1	☺ 1	◀ 17	!	1	A	Q	a	q
2	2	● 2	↕ 18	"	2	B	R	b	r
3	3	♥ 3	!! 19	#	3	C	S	c	s
4	4	♦ 4	¶ 20	\$	4	D	T	d	t
5	5	♣ 5	§ 21	%	5	E	U	e	ú
6	6	♠ 6	■ 22	&	6	F	V	f	v
7	7	• 7	↕ 23	'	7	G	W	g	w
8	8	• DEL ←	↑ 24	(8	H	X	h	x
9	9	○ ↕	↓ 25)	9	I	Y	i	y
10	A	○ 10	→ 26	*	:	J	Z	j	z
11	B	♂ 11	← 27	+	;	K	[k	{
12	C	♀ 12	└ 28	,	<	L	\	l	
13	D	♪ ↕	↔ 29	-	=	M]	m	}
14	E	♪ 14	▲ 30	.	>	N	^	n	~
15	F	☀ 15	▼ 31	/	?	O	—	o	△ 127

		DEC	128	144	160	176	192	208	224	240
		HEX	8	9	A	B	C	D	E	F
0	0	Ç	É	á	⌚	┌	∥	α	≡	
		128	144	160	176	192	208	224	240	
1	1	ü	æ	í	⌚	┌	┘	ß	±	
		129	145	161	177	193	209	225	241	
2	2	é	Æ	ó	⌚	┌	π	Γ	≥	
		130	146	162	178	194	210	226	242	
3	3	â	ô	ú	┌	┘	∥	π	≤	
		131	147	163	179	195	211	227	243	
4	4	ä	ö	ñ	┌	-	┘	Σ	∫	
		132	148	164	180	196	212	228	244	
5	5	à	ò	Ñ	┌	┘	ƒ	σ	∫	
		133	149	165	181	197	213	229	245	
6	6	å	û	ä	┌	┘	π	μ	÷	
		134	150	166	182	198	214	230	246	
7	7	ç	ù	ö	┌	┘	π	τ	≈	
		135	151	167	183	199	215	231	247	
8	8	ê	ÿ	¿	┌	┘	┘	♀	°	
		136	152	168	184	200	216	232	248	
9	9	ë	Ö	┌	┌	┘	┘	Θ	•	
		137	153	169	185	201	217	233	249	
10	A	è	Ü	┌	┌	┘	┘	Ω	•	
		138	154		186	202	218	234	250	
11	B	ï	ç	½	┌	┘	█	δ	√	
		139	155	171	187	203	219	235	251	
12	C	î	£	¼	┌	┘	█	∞	ⁿ	
		140	156	172	188	204	220	236	252	
13	D	ì	¥	ì	┌	=	┘	∅	²	
		141	157	173	189	205	221	237	253	
14	E	Ä	℥	<<	┌	┘	┘	€	¡	
		142	158	174	190	206	222	238	254	
15	F	Å	ƒ	>>	┌	┘	█	∩		
		143	159	175	191	207	223	239	BLANK	

Find the key.

The key is given in the bottom section of the table entry.

Where the table entry is not subdivided into boxes, just press the key on the keyboard that is marked with the character you want (while holding down the **[Shift]** key if you want either an upper case letter or the upper character written on the keytop).

If the bottom box contains a number, hold down the **[Alt]** key and type the number using the keys of the keypad and release the **[Alt]** key.

C.2 To find the internal code for a particular character

Find the internal code by adding together the appropriate number at the top of the column containing the character and the appropriate number at the beginning of its row.

If you want the internal code as a decimal number, add the outer pair of numbers. For example, the 'decimal' internal code for the character 'K' is $64 + 11$, that is, 75.

If you want the internal code as a hexadecimal number, add the inner pair of numbers by replacing the second character of the hexadecimal number at the top of the column with the character at the beginning of the row. For example, the 'hexadecimal' internal code for the character 'K' is $40 + B$, that is, 4B.

Note:

The character set given here is the full 8-bit character set. The section of the table on the lefthand page gives the characters that are available when a 7-bit character set is being used.

Appendix D:

INTERFACING TO AMSTRAD PPC HARDWARE

This appendix lists some fundamental information about the AMSTRAD PPC. The first section lists the codes to which the display responds; the second section lists the codes that the various keys on the keyboard generate. These are the codes used by programs to display information on the screen and to interpret what you type at the keyboard correctly. Tabulated at the end of the appendix are the device names used by MS-DOS for the input and output devices of your PPC.

For most users, the information given here is only of use when you are installing a new program. The installation program you run will sometimes ask you to type in details of the code that produces a particular effect, such as clearing the screen, and you can find that code in the tables below. More often though, it will ask you to press a particular key, so that it can learn in that way what the code is; or it will ask you which standard system your PPC emulates. This is also detailed below.

These codes are also of interest to programmers developing programs for the AMSTRAD PPC, whether in assembler or in higher level languages. The screen control codes detailed here can be used in print statements to produce particular effects on the screen, as the programming guide for the language which you are using will explain.

It should be noted that any references to colour apply only when an external monitor is being used with the PPC.

D.1 Screen codes

The standard screen handling is a very simple Teletype mode.

Characters with internal codes in the range 32..55 (&H20..FF in the hexadecimal notation) 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 (&H00...1F) are treated as control codes as follows:

7 (&H07)	B E L (Bell): Sounds a bleep
8 (&H08)	B S (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 (&H0A) L F (Line feed): Moves the cursor down one line, scrolling the screen up if necessary.
- 13 (&H0D) C R (Carriage return): Moves the cursor to the column furthest to the left on the present row.
- 27 (&H1B) E s c (Escape): Introduces an Escape sequence.

If the CONFIG.SYS file (see appendix E, Section E.3) is changed to include the command:

DEVICE=ANSI.SYS

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

- E s c [nA 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.
- E s c [nB 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.
- E s c [nC 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.
- E s c [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.
- E s c [n;mf Move the cursor to row n, column m. If n or m is omitted, the value 1 is assumed.
- E s c [n;mH Move the cursor to row n, column m. If n or m is omitted, the value 1 is assumed.
- E s c [=nh (E s c [?7h) Set screen width and type (see Table 1, below)
- E s c [2J Clear the screen and return the cursor to its Home position.
- E s c [K Erase to the end of the line.
- E s c [=nl (E s c [?7l) Reset screen width and type (see Table 1, below)
- E s c n;...;km Set graphics parameters (see Table 2, below).
- E s c [6n Report current cursor position to the system.
- E s c [standard-internal-code;parameterf;parameter...Jp
where parameter is a decimal number or a string
 Map standard internal code associated with a key onto another code or sequence of codes, thereby redefining the keyboard.
- E s c [n;mR Specify current cursor position as row n, column m and report this position to the system.
- E s c [s Save the cursor position.
- E s c [u Restore the cursor position.

Notes:

- i) Use of the screen is limited to a viewport which may be all or part of the screen. Row and column numbers should be given relative to the top lefthand corner of the viewport.
- ii) Operations that erase text only affect characters within the current viewport. However, scrolling the viewport can affect text outside the present viewport.

Table 1: Screen width and type

Ø	40 x 25 black and white
1	40 x 25 colour
2	80 x 25 black and white
3	80 x 25 colour
4	320 x 200 colour
5	320 x 200 black and white
6	640 x 200 black and white
7	wrap at end of line

Table 2: Graphics parameters

Ø	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
3Ø	Black foreground
31	Red foreground
32	Green foreground
33	Yellow (brown) foreground
34	Blue foreground
35	Magenta foreground
36	Cyan foreground
37	White foreground
4Ø	Black background
41	Red background
42	Green background
43	Yellow (brown) background
44	Blue background
45	Magenta background
46	Cyan background
47	White background
48	Subscript
49	Superscript

D.2 The keyboard

Most installation programs obtain the information they need about the keyboard by asking you to press specific keys at specific stages in the installation program. However, you may be asked to quote key codes instead and so these are listed here.

Notes:

It is possible to set up the function keys of your PPC to generate the strings of codes used, for example, by a word processor as an instruction to re-lay a paragraph. How to do this is outside the scope of this manual, but details should be obtainable from reference manuals on the operating system you will be using to run the program.

The key tokens given here are the ones that apply by default. After KEYB (see appendix E, Section E.2) has been run the keys produce the symbols as defined on the keyboard so different key tokens are produced.

Keycodes and their translation

The Keycode translation table is as follows (all values are hexadecimal).

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	←	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

Key Code	Key Name	Normal	ALT	CTRL	SHIFT	Num Lock
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	Lft SHIFT	Ignored	Ignored	Ignored	Ignored	N/A
2B	\ and	2B5C	2BF0	2B1C	2B7C	N/A
2C	Z	2C7A	2C00	2C1A	2C5A	N/A
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	Rgt 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

Key Code	Key Name	Normal	ALT	CTRL	SHIFT	Num Lock
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	NUMLOCK	Ignored	Ignored	PAUSE	Ignored	N/A
46	SCROLLLOCK	Ignored	Ignored	Ignored	Ignored	N/A
47	KEY PAD 7	4700	Ignored	7700	} Note 1	4737
48	KEY PAD 8	4800	Ignored	8D00		4838
49	KEY PAD 9	4900	Ignored	8400		4939
4A	KEY PAD -	4A2D	Ignored	8E00		4A2D
4B	KEY PAD 4	4B00	Ignored	7300		4B34
4C	KEY PAD 5	Ignored	Ignored	8F00		4C35
4D	KEY PAD 6	4D00	Ignored	7400		4D36
4E	KEY PAD +	4E2B	Ignored	9000		4E2B
4F	KEY PAD 1	4F00	Ignored	7500		4F31
50	KEY PAD 2	5000	Ignored	9100		5032
51	KEY PAD 3	5100	Ignored	7600		5133
52	KEY PAD 0	5200	Ignored	9200		5230
53	KEY PAD .	5300	Ignored	9300		532E
54	Alt PrScr	Ignored	Sys Req	Ignored		Ignored
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

Extra Keys and their tokens

E0,1C	ENTER	E00D	A600	E00A	E00D	N/A
E0,1D	RightCTRL	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

The following set of keys are the dedicated cursor keys. They produce different sequences of key codes depending on state of the [SHIFT] key and the [Num Lock] light. In the base state and the state with the [SHIFT] key down plus the [Num Lock] light on, the key sequences consist of the keycodes expected from the corresponding key on the numeric keypad (with the [Num Lock] off) preceded by the extra key keycode (that is, E0h). With only the [Num Lock] light on the key sequence consists of the base sequence preceded by E0h,2Ah so there now are four keycodes in the sequence. The last variation is if the [SHIFT] key only is down. In this case the sequence as shown in the base case is preceded by E0h,AAh and again there are four keycodes in the sequence.

E0,46	Prt Scrn	Ignored	Ignored	Ignored	Ignored	N/A
E0,47	HOME	47E0	9700	77E0	47E0	N/A
E0,48	UP ARROW	48E0	9800	8DE0	48E0	N/A
E0,49	PAGE UP	49E0	9900	77E0	49E0	N/A
E0,4B	LT ARROW	4BE0	9B00	73E0	4BE0	N/A
E0,4D	RT ARROW	4DE0	9D00	74E0	4DE0	N/A
E0,4F	END	4FE0	9F00	75E0	4FE0	N/A
E0,50	DN ARROW	50E0	A000	91E0	50E0	N/A
E0,51	PAGE DN	51E0	A100	76E0	51E0	N/A
E0,52	INSERT	52E0	A200	92E0	52E0	N/A
E0,53	DELETE	53E0	A300	93E0	53E0	N/A

The last set of keys are classed as miscellaneous.

E0,37	CTRL or SHIFT Print Screen
E0,2A,E0,37	Print Screen
E0,AA,E0,35	Shift / on keypad
E1,1D,45	Pause

Note 1:

With the [Num Lock] light on and the [SHIFT] key down, these keys produce the token from the Normal column above. When the [Num Lock] light is out and the [SHIFT] key down then the token produced is the one from the Num Lock column.

Note 2:

The reset sequence is invoked by pressing [Ctrl][Alt] and [Del].

Key tokens shown above, greater than 8400, are produced but are ignored or translated by the system. For further information consult the Technical Reference Manual [section 2.3.11]

The key code marked '*' is handled by ROS but cannot be produced by the US (101 key) version of the keyboard.

D.3 Device names

DEVICE NAMES UNDER MS-DOS

CON	Both the keyboard and the screen
AUX or COM 1	The Serial Interface
COM 2	Used by the internal modem in the PPC 640 machines and reserved for use on the PPC 512 if a second communications port is added.
PRN or LPT 1	The Parallel Printer port.

Appendix E:

EXPANDING YOUR AMSTRAD PPC

This appendix describes how to go about expanding your AMSTRAD PPC by:

- adding an external monitor
- adding a printer or a comms link

It gives both the technical aspects of choosing how to expand your system and explains the steps to incorporating these additions in your system. It also explains how to reconfigure your system software to accommodate the changes you have made.

E.1 Adding hardware

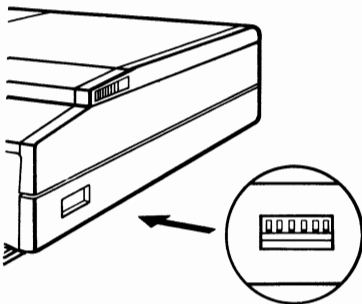
E.1.1 Adding an external monitor

The Video connector on the back of the PPC is a 9-pin D-type socket. An AMSTRAD PC-MD, PC-CD, or PC-ECD monitor, or any IBM-PC or IBM-PC compatible monitor may be connected to this socket.

Note:

If an AMSTRAD monitor is used in this way the PPC must be powered from the monitor's 12v D.C. output.

Monitor control switch settings.



Normal dip switch settings (↑ = up, ↓ = down)

Switch No	1	2	3	4	5
MDA (Mono)	↓	↑	↓	↓	↓
CGA (Colour)	↓	↓	↓	↓	↑

Note:

The mode must be selected to suit the external monitor, when used. The LCD works in both CGA and MDA modes.

Most of the settings are default modes, which determine how the PPC behaves at switch-on. They can be overridden by software commands if desired. However, the settings which concern whether a colour or monochrome external VDU is connected to the PPC **MUST** be made correctly if the VDU is to work properly.

Switches 1 to 3 control the internal hardware.

Switch 1 determines whether the internal LCD or the external VDU is initially selected. The down position selects LCD, up selects VDU. During the internal checks which are carried out each time the PPC is reset, the message:

```
P l e a s e   w a i t  
* * * * *
```

is always shown on the LCD, but all subsequent output will be directed to the device selected by dipswitch no.1, until the display is toggled by the software switch (press **[Scroll Lock]** and **[Ctrl]**).

Note:

No warning is given if an external monitor is selected but not actually connected to the PPC, although the CRT indicator on top of the PPC will be lit. The LCD screen will simply remain blank. The output is not lost; it will appear as soon as the external device is connected or the Scroll Lock and Ctrl keys are used to toggle the output to the LCD.

Switch 2 must be down, to select CGA, when a colour VDU is in use; up, to select MDA, for a monochrome VDU. In MDA mode, only an 80 column screen width is available.

Switch 3 must always be down unless an external display adaptor has been connected.

Switches 4 and 5 must be down (select MDA) when a monochrome VDU is connected. They are used to select the default screen width when a color VDU is used, but then the settings can be overridden by using the DOS command MODE.

Note:

Changing the dipswitch settings has no effect until the PPC is reset.

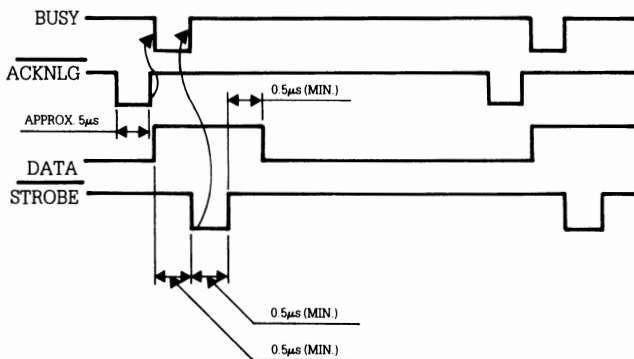
E.1.2 Making use of the Parallel Printer connector

Important: When connecting a printer a UL/FCC approved cable must be used.

The Parallel Printer connector on the back of the PPC is a 25-pin 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 DMP3000 which is compatible with both EPSON FX80 and IBM Personal Computer Graphics Printer.

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

Computer Connector Pin	Printer Connector Pin	Return Pin	Designation	I/O	Notes
1	1	19	<u>STROBE</u>	IN	The signal level is normally high; it is taken low to send data. The pulse width must be more than 0.5 μ s 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	<u>ACKNOWLEDGE</u>	OUT	Pin taken low to indicate that your printer is ready to receive further data. Approximately 0.5 μ S 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.
14	14	-	AUTOFEED	IN	Pin low means paper to be fed one line after printing.
-	15	-	NC		Not used.
-	16	-	OV		Logic GND
-	17	-	CHASSIS GND		Printer chassis GND.
-	18	-	NC		Not used.
19...25	19...25	-	GND		Twisted-pair Return signal GND.
-	26...30				
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 0.5 μ s 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 used.
-	35	-	5V		Pulled up to +5V through a 4.7k Ω resistance.
17	36	-	<u>SLCT IN</u>		Data entry to printer only possible when pin low. Pin taken high to indicate printer off-line.



To make the connection between the printer/plotter and your PPC, you will need an **AMSTRAD PL-2** or equivalent **IBM PC to Centronics** printer cable. This cable should have a 'male' plug on one end to fit the 'female' connector on your **AMSTRAD PC**: the type of plug it should have on the other end depends on whether the printer has a 'male' or a 'female' connector. Your dealer should be able to help you here.

The 'male' end of this cable should be plugged into the Parallel Printer port on the back of your PC and the other end into 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 Printer 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. The **AMSTRAD DMP3000**, for example, is factory set to the European version of the IBM character set, but has switches on it that change this to one of three other character sets.

E.1.3 Making use of the Serial Interface

The Serial Interface connector on the back of the **AMSTRAD PPC** is a 25-pin D-type connector, which provides a standard **RS232C** interface and so can be used to connect any comms link or printer that uses the standard **RS232C** serial interface. This gives you a wide choice of modems, comms links and printers.

If you or your dealer are not sure whether a comms link or a printer is suitable, compare the pin-out given for the device with the pin-out given below.

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

To make the connection between the device and your PPC, you will need a suitable cable. The type of cable you need depends on what you want to connect to the Serial Interface:

- * **To connect to a Modem or a 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**

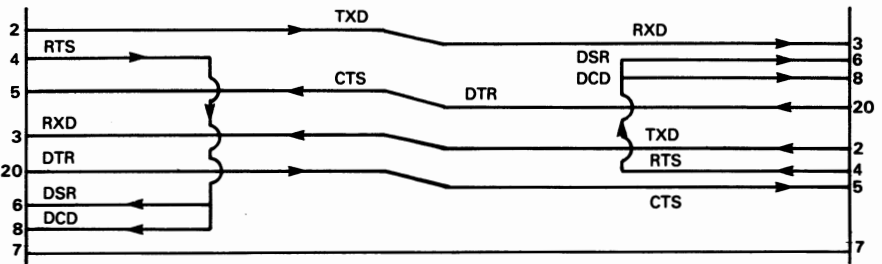
IMPORTANT: WHEN CONNECTING A DEVICE A UL/FCC APPROVED CABLE MUST BE USED.

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** (that is, Pin 2 is an output pin), then the device is wired a terminal and you need a Null-modem cable
- if Pin 2 is used to **receive** (that is, Pin 2 is an input pin), then the device is wired as a modem and you need a one-to-one cable

Your cable should have a 'female' plug on one end to fit the 'male' connector on your AMSTRAD 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 a 'female' connector. Your dealer should be able to help you here.

The 'female' end of this cable should be plugged into the Serial Interface port on the back of your PPC 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 Interface 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 expect to use the Serial Interface to link your PPC with a range of external devices, make the settings by using a MODE command (see section 2.7.1). The command line could usefully be included in the Batch file to run a particular program or in the AUTOEXEC.BAT file on a particular Startup disk.

E.2. Configuring your MS.DOS system software.

MS-DOS Version 3.3 is initially configured to suit a wide range of 16-bit microcomputers. On the AMSTRAD PPC, this configuration is modified by settings in the CONFIG.SYS file in the Root directory of the PPC SYSTEM Disk. This file is automatically processed by MS-DOS as part of the Startup procedure. Further changes to the configuration can be made by:

- * **modifying the CONFIG.SYS file**
- * **running a program called ANSI.COM**
- * **loading a different keyboard program**

E.2.1. Changing the CONFIG.SYS file

The commonest reasons for changing the CONFIG.SYS file are:

- to increase the number of disk buffers if your work involves a lot of disk accessing
- to increase the number of files that can be open simultaneously

The CONFIG.SYS file draws on special commands, full details of which can be found in the AMSTRAD PPC Technical Reference (available from AMSTRAD). The ones covered here are the ones that don't need special technical knowledge to use. These are, in any case, the ones you are most likely to need.

BREAK=ON|OFF

This does a similar job to the MS-DOS BREAK command, increasing the number of times MS-DOS checks whether Ctrl-C has been pressed when BREAK is set to ON and limiting the checking to when the program is using the keyboard or the screen when BREAK is set to OFF.

BUFFERS=n

This command sets the number of sector buffers available when data is transferred to and from disk. A word-processor program will typically need 10...20. You will also need a large number of buffers if you have many subdirectories on any disks, but don't set a number higher than you need because each buffer takes 0.5K of RAM. The AMSTRAD PC is supplied with 5 buffers; the default number is 2.

COUNTRY = nnn	This command tells MS-DOS to use the time date and currency convention of the country specified by nnn. nnn is the three-digit number used to represent the country in the International Dialling Codes: so the UK is 0 4 4 and the US is 0 0 1.
FCBS = x, y	This command sets the number of files opened by File Control Blocks (FCBs) that can be open at any one time (x) and the number (y) that are protected against being closed automatically by MS-DOS if an application tries to open further files. The possible range for x is 0...255 (default 4)
FILES = n	This command sets the number of file handles that can be in use simultaneously. The possible range is 8...255, though 20 is the maximum in practice. The AMSTRAD PC is supplied with a limit of 20 file handles; the default number of file handles is 8.

Loading a different keyboard program

KEYB xx

Keyboard layout varies from country to country, because of the different needs of the languages spoken there.

MS-DOS is initially set up to work with an American style keyboard. However, it can be configured to work with a different national keyboard by running a special program. This program has the name **KEYB xx**, where **xx** indicates which national keyboard layout will be set, as indicated in the following table:

USA	KEYB US
British	KEYB UK
German	KEYB GR
French	KEYB FR
Spanish	KEYB SP
Italian	KEYB IT

The AMSTRAD PPC is supplied with an **AUTOEXEC.BAT** which invokes the correct **KEYB xx** program for the country in which it is sold.

Appendix F:

REFERENCE MATERIAL ON THE AMSTRAD PPC

This manual mainly covers the everyday use of your AMSTRAD PPC. It illustrates how to use your PPC to run AMSTRAD or IBM compatible software which you have bought, plus the use of DOS commands to organise and maintain your program and data files. It also contains a brief introduction to communications and instructions for the use of the MIRROR II software provided with the AMSTRAD PPC640.

If you want to run non-compatible applications and write programs in other languages, to run under either MS-DOS or other operating systems, you may require a more detailed knowledge of your AMSTRAD PPC. This is beyond the scope of this user guide. However, to help you acquire this knowledge, this appendix summarises the information you will need to know and the technical manuals you will need to buy.

Your AMSTRAD PPC system consists of hardware and software.

The hardware is the computer itself - the keyboard, the monitor, etc. This is useless without software to control it. The software comes in various types from very low level software that controls the hardware to application programs such as word processors or spreadsheets.

The hardware and the various levels of software are described in a variety of technical manuals. These are listed below.

Guides to the software

The various levels of software you use on your AMSTRAD PPC are:

- * **Application programs such as PPC ORGANIZER, providing facilities such as spreadsheet, word processor, etc**
- * **An operating system, such as MS-DOS**
- * **The very low level software which controls the physical hardware. This is variously known as the Resident Operating System (ROS), the ROM or the firmware.**

Applications programs:

PPC ORGANIZER is supplied with full documentation.

For a detailed description of MIRROR II, see:

MIRROR Data Communications for the IBM PC and compatibles.

available from Softklone Distributing Corp.

Other applications programs make certain common demands of your PPC, such as reading and writing files, displaying information etc. However, all the information you need to run the programs should be given in their own user guides.

The operating system:

- handles all the common functions which the applications need, and provides housekeeping facilities through the Command Interpreter (The functions provided by the Command Interpreter are described in section 2 of this manual.)

For information on the structure of MS-DOS, reading and writing files and disks, control of the console and other peripherals and full details of utilities such as DEBUG see:

MS-DOS Programmers' Reference Manual - available from Microsoft

The AMSTRAD PPC hardware

Using the hardware through the operating system can be slow, and many commercially available software packages use the firmware or hardware directly.

For information on the hardware and firmware of the AMSTRAD PPC:

AMSTRAD PPC Technical Reference - available from AMSTRAD

The Hardware sections will provide details of, for example, the full range of functions of which the screen is capable and how to use them.

The Firmware sections describe the low level interface to the hardware, and will indicate how to call the functions described in the hardware specification.

The Technical Reference also describes the full range of CONFIG.SYS commands.

Add-on hardware

You can add hardware to your PPC by simply plugging the add-on hardware (such as a printer), or its 'interface', into the appropriate socket on the rear of the AMSTRAD PPC. You should refer to the fitting instructions which accompany the add-on, and read them in conjunction with Appendix E.

If you need to use add-on hardware from your own programs, or install software you have bought to use add-on hardware, you should refer to the user manual accompanying the add-on hardware. It may be that there is a hardware specification available for your add-on. If there is, use it as an extension of the AMSTRAD PPC's hardware specification.

Appendix G:

TROUBLESHOOTING

This appendix looks at what to do and what might have gone wrong when your PPC or your program doesn't work in the way you expect. If you can't find the solution to your problem here, consult your dealer.

Note:

If your PPC cannot accept a character it beeps. If you don't hear any beep, adjust the volume control on top of the PPC.

G.1 Trouble during Startup or when resetting your PPC

*** If nothing happens at all:**

Check if the PPC will work on its internal batteries, or another approved power source.

Check that the power socket is working by plugging in and switching on a lamp you know is working.

Check that the plug on the A.C. adaptor is correctly wired and that the power fuse has not blown.

If none of these actions shows you where the problem is, consult your dealer.

*** The PPC beeps and asks you to insert a System disk when you insert your Startup disk**

Check that you inserted the right disk and then reset the machine. If it beeps again, the disk it is trying to read either is damaged or does not have any operating system software on it. Try another Startup disk: if you don't have another Startup disk or if your PPC 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 the AMSTRAD PPC.

As soon as possible after this failure, use MS-DOS's CHKDSK command (see section 2.6.3) to find out whether the disk that failed was damaged.

*** Software is read into your PPC's memory but the machine dies.**

Check that you inserted the right disk and then reset the machine. If your PPC dies again, the software on your disk has been corrupted. Try another Startup disk: if you don't have another Startup disk, consult your dealer.

- * **The screen display when all the software has been read in is plausible but isn't 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 your PPC, using another Startup disk.

G.2 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
 - the file is not on the disk you are currently accessing
 - 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 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 isn't 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 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 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 2. 4.2.1. 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 G.6 - 9 of this appendix 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 PPC 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 3.3.

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 don't help you identify what the problem is, reset your PPC (See section 1.7.3) and then try again. If the same failure happens again and it's 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.

G.3 Trouble with the keyboard

- * **If your PPC fails to respond to what you type:**

If the cursor isn't flashing, something in your program has caused your PPC to 'die'. Put your Startup disk into Drive A and reset your machine by holding down [Ctrl] and [Alt] and pressing [Del]. If your PPC 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 PPC Startup procedure (see section 1.7.1).

If your PPC 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 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 don't produce the actions/characters you expect**

You may have used the KEYB utility from a different DOS-compatible operating system. You must use the operating systems supplied with your PPC to get correct keyboard operation.

G.4 Trouble with a disk drive

If your PPC frequently reports that it can't 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 the AMSTRAD PPC. Your dealer will be able to advise you on which one to use.

If cleaning the disk drive doesn't 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. Don't limp on with a defective drive - you could easily damage your disks - and don't try using a two-drive machine as a single-drive system by disconnecting the defective drive. If you do, (i) you will invalidate the warranty on your PPC, and (ii) you will find that the initial system check will fail and you won't be able to use the machine anyway.

There is no alternative to getting a defective disk drive fixed.

G.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 doesn't 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 PPC 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 manual to see whether the printer can print different numbers of characters per line or lines per inch. The AMSTRAD DMP1, 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, for example, 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 AMSTRAD DMP3000).

G.6 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 occurred: the options are:

Bad call format error	Non-DOS disk error
Bad command error	No paper error
Bad media type	Not ready error
Bad unit error	Read fault error
Data error	Sector not found error
FCB unavailable	Seek error
General failure	Sharing violation
Invalid disk change	Write fault error
I/O error	Write protect error
Lock violation	

Type **[A]** to abandon the current command or program; type **[R]** to tell the operating system to try the action again; type **[I]** to tell it to move on to the next part of the task.

Note:

The FCB error can usually be cured by changing the FCBS command in your CONFIG.SYS file.

G.7 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.

```
Access denied
```

Your disk is probably write-protected.

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

You have just told MS-DOS to erase all the files in a directory. Type Y if you want all these files deleted; type N to abandon this command.

Attempted write-protect violation

The disk you are trying to format is write-protected.

Bad command or filename

The command cannot find the program you asked to run. Check that you typed its name correctly or the file is not in the directory you specified.

Bad or missing filename

The CONFIG.SYS file contains an invalid device name. Check the DEVICE command in the CONFIG.SYS file (see appendix E, section E.2).

Bad or missing Command Interpreter

or

Cannot Load CCP

The operating system cannot find the COMMAND.COM file. Either reset your machine or copy the COMMAND.COM file from your backup operating system disk onto your Startup disk.

Batch file *file-name* missing retry (Y/N)?

Your computer is in the middle of running a Batch file but the disk containing the Batch file has been removed from the drive. Re-insert the disk and then press any character key.

Cannot do binary read from a device

The /B option of the COPY command cannot be used where an Input Device is the source of the file. Either remove the /B option or replace it with /A

Cannot open filename

The operating system cannot find the specified file. Check your command.

Cannot recover . entry

The working directory is defective.

CHDIR .. failed, trying alternate method

The operating system was unable to move directly to a parent directory. It will try to get to the directory by starting at the root directory and working down the directory tree.

COM port does not exist

You have specified an invalid COM port.

Content of destination lost before copy

file to be used as a source file to the COPY command has been overwritten prior to the copy being made.

Convert lost chains to files (Y/N)?

If you type Y, CHKDSK will recover the lost blocks it found while checking the disk and store them in files. If you type N, the lost blocks are deleted.

Copy not completed

DISKCOPY was unable to copy the entire disk.

Corrections will not be written to disk

CHKDSK has found errors on the disk but won't fix any of these as you didn't select the / F option.

Directory is joined

The command cannot process directories that have been joined with the JOIN command.

Directory is totally empty, no . or ..

The directory does not contain entries for the working and the parent directory. Delete the directory and then recreate it.

Disk error reading FAT n

or

Disk error writing FAT n

One of your File Allocation Tables is damaged. You are advised to copy your files onto another disk.

Disk full. Edits lost

There was insufficient room on the disk for EDLIN to save your edited file. You will have to edit the file again.

Disks must be the same size

DISKCOPY can only copy a disk if the target disk can be given the same format as the source disk. You can, however, use the COPY command to copy the files across one by one.

Disk Read/Write Error

Your disk has been corrupted or damaged. You are advised to copy what files you can onto another disk and then try reformatting the disk.

Disk unsuitable for system disk

The disk is damaged in the area set aside for the system files. The disk can't be used as a startup disk but can be used for storing programs and data.

Divide error

or

Divide overflow

The hardware divide overflow flag has been set. The probable reason for this is that your program file has become corrupted, causing it to execute random code.

Duplicate filename

file with the filename you chose already exists.

Entry error

The last EDLIN command you typed contains a syntax error. Redo the command.

Entry has a bad attributellinksize

CHKDSK has found a damaged part of the directory tree.

ERROR: Diskette is full

There is insufficient room on the disk to save your file.

Error in .EXE file

The .EXE file you asked the operating system to load has an invalid internal format

Errors on list device indicate that it may be off-line. Please check it

Your printer isn't turned on.

Error writing to device

You tried to send too much data to your Output Device.

EXEC failure

The operating system either found an error while reading an internal command or the number of open file handles has been set too low. Check the FILES command in the CONFIG.SYS file (see appendix E, Section E.2).

fc: out of memory

COMP does not have enough memory to perform the comparison.

File allocation table bad

Your disk may be damaged. Run CHKDSK with the / F option to fix the disk.

File creation error

Either a file already exists with the name you chose and it cannot be replaced (eg. because it is a Read-Only file) or your disk may be damaged.

File not erased

The file you told the operating system to erase is write-protected.

File not found

MS-DOS cannot find the file you specified. Check your command and that the file you want is in the directory you specified.

FIRST diskette bad or incompatible

DISKCOMP failed to recognise the format of the source disk. Run CHKDSK to find out what the problem is.

For cannot be nested

You may not nest FOR commands in a Batch file.

Format not supported on drive n:

You cannot use the FORMAT command to format this drive.

Illegal device name

The operating system does not recognise this device name.

Illegal wildcard

The file name template you gave is invalid.

Incompatible system size

The system files cannot be copied because they need more space than is available on the destination disk.

Incorrect DOS version

You are using a version of MS-DOS that doesn't support this command.

Incorrect number of parameters

or

Invalid number of parameters

You put either too many or too few parameters in the command line.

Insert diskette with batch file and press any key when ready

Your computer is in the middle of running a Batch file but the disk containing the Batch file has been removed from the drive. Re-insert the disk and then press any character key.

Insert DOS disk in drive n: and press RETURN when ready

The drive does not contain the MS-DOS system files you have asked to copy by selecting the /S option of the FORMAT command. Insert your MS-DOS startup disk in the specified drive and then press [↵]

Insert FIRST diskette in drive n:

Insert the disk you want DISKCOMP to compare another disk against. Press any character key when ready.

Insert new diskette for drive n: and press RETURN when ready

Insert the disk you want to format. Press [↵] when ready.

Insert SECOND diskette in drive n:

Insert the disk you want DISKCOMP to compare with your original disk. Press any character key when ready.

Insert target diskette into drive *n*:

Insert the disk you want DISKCOPY to store the copy on in the specified drive. Press any character key when ready.

Insufficient disk space

Your disk is too full to carry out the specified task.

Insufficient memory

Your machine has too little memory available to carry out the specified task.

Intermediate file error during pipe

An error has occurred in one of the temporary files created when information is piped from one program to another.

Invalid COMMAND.COM

The program you have just run used up almost all the available memory, overwriting the transient part of the COMMAND.COM file. The COMMAND.COM file has to be re-read from disk.

Invalid device

The device specified was not CON, NUL, AUX or PRN.

Invalid directory

or

Invalid sub-directory entry

The directory you specified either doesn't exist or is invalid. Check that you entered the directory name correctly.

Invalid drive in search path

or

Illegal drive in Path specification

A drive specified in the search path does not exist.

Invalid option

or

Invalid parameter

One of the options you specified is wrong.

Invalid path, not directory

or directory not empty

RMDIR was either unable to find the directory you wanted to delete or the directory still contained files and/or subdirectories.

Invalid working directory

Your disk is damaged.

Line too long

Searching and replacing in EDLIN caused a line to expand beyond 253 characters. Divide the line in two and then retry the Replace command.

LPTn: not redirected

Usually means that you have just specified the settings of this printer device and you haven't told MS-DOS to redirect the printer output to a Serial port. Occasionally, it will indicate that the MODE command was unable to redirect the parallel printer port. Check that you specified the right printer options.

Memory allocation error

Reset your PPC. If this error persists, make a new startup disk.

No free file handles

Reset your PPC. If this error persists, increase the number of open file handles specified by the FILES command in the CONFIG.SYS file (see appendix E, section E.3).

Non-system disk or disk error

Either the disk is not a startup disk or the disk is damaged. Try another startup disk.

No room in directory for file

EDLIN was unable to save your edited file because the directory is full. Try saving the file in one of the other directories on the disk.

No system on default drive

SYS cannot copy the system files to the target disk because the disk in the default drive does not contain these files. Replace the disk with your MS-DOS startup disk and try again.

Not enough memory

There is insufficient memory to run the command.

Out of environment space

There is no room for further environment strings (see the SET command).

Parameters not compatible

The options you have selected can't be used together.

Probable non-DOS disk. Continue (Y/N)?

The disk you are using is not recognised by MS-DOS. Do not continue processing with CHKDSK.

Processing cannot continue

Insufficient memory for the current task.

Read error in filename

The operating system could not read this file.

SECOND diskette bad or incompatible

DISKCOMP could not recognise the format of the target disk for the comparison. Use CHKDSK to help you find out what the problem is.

Specified drive does not exist, or is non-removable

The command cannot work with the drive you specified (for example, because it doesn't exist).

Syntax error

Check you have typed the command correctly.

Too many files open

Insufficient file handles to open another file. Increase the number of open file handles specified by the FILES command in the CONFIG.SYS file (see appendix E, section E.3).

Track 0 bad - disk unusable

The disk is damaged in track 0, making it impossible to format. Use another disk.

Unable to create directory

You probably specified a name that is already used either by another sub-directory of the same parent directory or by a file in this directory. Alternatively, the disk may be full.

Unable to MOVE files use COPY

Your RENAME command gave a different drive in specifying the new file name to that used to specify the old file name. The file was not renamed. Re-do your command line.

Unrecoverable error in directory. Convert directory to file (Y/N)?

If you type Y, you may later be able to fix the damaged directory yourself or you decide to delete it.

usage: fc [/a] [/b] [/c] [/l] [/**l** n] [/**w**] [t]
[/**n**] [/**NNNN**] file1 file2

One of the options you specified in your FC command was invalid.

Warning - directory full

The directory is full and so your file cannot be saved. Delete some files in the root directory.

Warning: Read error in EXE file

The amount read was less than the size of the header. This is a warning message only.

G.8 Hardware error messages

Non-fatal errors

Check Keyboard

One of the keys on the keyboard is either being held down or it has become stuck down. Check that nothing is resting on the keys or jammed between them.

Fatal errors

The following messages are displayed when internal checking has revealed an error that would stop your PPC from working properly. The message tells you the area of the hardware in which the error was found.

Your PPC has to be switched off and then switched on again before it can be used further. If the error persists, consult your dealer.

Error: Faulty direct memory access controller
Error: Faulty floppy disk controller or disk drive
Error: Faulty interrupt controller
Error: Faulty interval timer
Error: Faulty memory (parity error)
Error: Faulty real time clock
Error: Faulty ROS checksum
Error: Faulty system printer port
Error: Faulty SYSTEM RAM
Error: Faulty system status register
Error: Faulty VDU controller
Error: Faulty VDU RAM

APPENDIX H. THE PC 2000 MODEM

1. U.S. STATUTORY REQUIREMENTS

2. COMMAND SET

3. REGISTERS

4. RESULT CODES

5. MODES SUPPORTED BY PC2000 MODEM

6. REQUEST FOR DATA MODEM CONNECTION TO BT PSTN

1. US STATUTORY REQUIREMENTS

The following information is legally required to be given and must be heeded by the user.

This equipment complies with part 68 of the FCC Rules. On the bottom of this equipment is a label that contains, among other information, the FCC Registration Number and Ringer Equivalence Number (REN) for this equipment. You must, upon request, provide this information to your telephone company.

The REN is useful to determine the quantity of devices you may connect to your telephone line and still have all of those devices ring when your telephone number is called. In most, but not all areas, the sum of the RENs of all devices connected to one line should not exceed five (5.0). To be certain of the number of devices you may connect to your line, as determined by the REN, you should contact your local telephone company to determine the maximum REN for your calling area.

If your telephone equipment causes harm to the telephone network, the Telephone Company may discontinue your service temporarily. If possible, they will notify in advance. But if advance notice isn't practical, you will be notified as soon as possible. You will be informed of your right to file a complaint with the FCC.

Your telephone company may make changes in its facilities, equipment, operations, or procedures that could affect the proper functioning of your equipment. If they do, you will be notified in advance to give you an opportunity to maintain uninterrupted telephone service.

If you experience trouble with this telephone equipment, please contact Amstrad National Service Center, 1915 Harrison Road, Longview, Texas 75604 for the address of your local Amstrad authorized dealer or Amstrad authorized Service center. The telephone company may ask that you disconnect this equipment from the network until the problem has been corrected or until you are sure that the equipment is not malfunctioning.

This equipment may not be used on coin service provided by the telephone company. Connection to party lines is subject to state tariffs.

The Ringer Equivalence No. is: 0.3B

INSTALLATION

This device is equipped with USOC RJ11C connectors.

2. HAYES-COMPATIBLE COMMAND SET

All commands must be preceded by AT except the command A/. However, a series of commands may be entered consecutively on one line so long as the series starts with AT and terminates with a [↵]. If a command series contains a command that forces entry to data-mode (such as ATA, ATD, ATO) any following commands will not be executed as commands from the host are not executed in data-mode (except for the escape sequence).

All commands must be terminated by [↵] except the commands A% and A/. The current value (ASCII) for [↵] is held in S3.

Commands may be entered in upper or lower case including the AT part. But note that serial protocol determination is most efficient with upper case for the AT part.

The buffer is 40 bytes (characters) long; if a command exceeds this length, the buffer will be cleared and an error message sent. (No part of the command will be executed). The AT does not take up any space in buffer.

Any character with an ASCII value of less than 32 or more than 127 (decimal) will not be entered into the buffer. Spaces may be used to make the command line more readable; they will take up space in the buffer, but they will not affect execution. Any character within the range 32-127 inclusive (decimal) that is not a valid command or parameter will produce an error message when execution is attempted.

If a command contains an error, any instructions within the command occurring before the error will be executed, but any commands occurring after the error will not be executed. An ERROR message will be issued to the host computer if result codes are enabled.

All numbers quoted are in decimal.

Default values refer to the power-up setting and are not necessarily the same as that set by a command with a zero or missing parameter.

COMMAND LIST

ATA Answer immediately, go on line, generate answer carrier and wait for incoming carrier. This would probably be used after a voice contact had been established to enter data mode. The remote end would enter ATD to take the line at its end and would be the originating modem.

- A / Repeat last command in buffer; buffer is cleared by a new AT or a power-on, making A / invalid. Usually used to redial last number. Used without AT and [←].
- A% Display the contents of the command buffer. This command does not destroy the buffer so an A / command will attempt to execute what is in the buffer.

ATBn BELL/CCITT mode select.

Range(n): 1,4,5,8,17,20,21,24 Default=5 (V22bis orig).

This command is used to change the communication mode of the modem. In the U.S., it will not normally be necessary at all. This is because the modem automatically selects a mode to use on the basis of the incoming AT command parameters. (See the section entitled 'Baud-rate and serial format determination from AT command'). When using a mode change command, the best place to put it is in the dial prefix string.

n=	1- Bell 103	300 bps Full duplex	: Originate.
	4- V22 bis,	1200 bps Full duplex	: Originate.
	5- V22 bis	2400 bps Full duplex	: Originate.
	8- Bell 212A	1200 bps Full duplex	: Originate.
	17- Bell 103	300 bps Full duplex	: Answer.
	20- V22 bis,	1200 bps Full duplex	: Answer.
	21- V22 bis	2400 bps Full duplex	: Answer.
	24- Bell 212A	1200 bps Full duplex	: Answer.

Notes

- (i) The value programmed with the ATB command (if suitable) is put into register S19 which is read-only. The value in S19 may be changed by the modem during the course of a connection e.g. an R within a dial command causes the modem to originate a call, but act as an answering modem. This would cause an originate value in S19 to be converted into an answer value.
- (ii) The modem will only allow selection of a mode which is compatible with the speed of the incoming ATB command. e.g. If the command is received at 2400 bps only ATB5 and ATB21 will be accepted- all other parameters will produce an ERROR message.

ATCn Carrier on/off.

Range(n): 0-1 Default=1

n=0- Transmitter off. Once this command has been entered, carrier will not turn on again until an ATZ (reset) or an ATC1 is issued.

ATXn Result code set. See result code list below.

Range(n): 0-255

Default= 1

n=0- Basic result code set.

n= 1 to 255- Extended result code set.

ATZ Terminate link and reset default parameters. This is equivalent to a full software reset and replaces ALL parameters with defaults.

[←] Backspace key deletes the last letter on the command line prior to pressing <RET> but will not delete AT or A/.

[←] Completion of the command line, execution will not proceed without this.

<missing numeric parameter>

A missing numeric parameter is assumed to be a parameter of 0. Thus, a command H has the same effect as H0 - Hangup (drop line).

, Pause for period (seconds) defined by S8. Generally used within dialling command to space digit groups.

; Return modem to command state after autodialling. This command is only valid within a dial string.

+++ The default ASCII value for the escape character is '+' (43D), but this can be altered by programming a new value into S2. Three consecutive escape characters is the escape sequence to command mode. Used to force the modem from online mode to command mode. This will have no effect unless it is preceded and followed by a <no character> guard time of a length determined by the value in S12. The escape command does not hang up the line, so it is possible to return to the line once any commands have been issued, using the the O command. Note that the three escape characters will be transmitted to the remote modem because the escape sequence to the command mode cannot be implemented until <guard-time> after they have been transmitted.

3. REGISTERS

NOTE

The range values shown are maximums. Some of the allowed range values for the registers are less than those shown due to practice restrictions for the particular mode/country.

S0 Autoanswer off/on;number of rings.

0-255 Default=0 Unit=Rings. R/W.

Number of rings allowed before modem autoanswers.

0-Modem never autoanswers.

In practice a value of 1 - 4 is probably optimal for auto-answering.

S1 Incoming ring count.

0-255 Default=0. Units=Rings. Read only.

Counts the number of incoming rings, and is cleared to 0 about 8 seconds after the last ring is sensed and only operates if register S0 is set to >0. This register is primarily for internal use.

S2 ASCII value of escape code.

0-127 Default=43. Units=Decimal R/W.

S3 ASCII value of carriage return or end of line character.

0-127 Default=13 Units=Decimal R/W.

S4 ASCII value of line feed character.

0-127 Default=10 Units=Decimal R/W.

S5 ASCII value of backspace character.

0-32,127 Default=8 Units=Decimal R/W.

S6 User-defined.

0-255 Default=0 Units=None R/W.

S7 Wait time for carrier after dialling or answering.

0-255 Default=55 Units=Seconds. R/W.

The value in here is loaded into a timer at the end of a dial command or after auto-answer. If no carrier is detected within the timeout period, the modem will drop the line and return a NO CARRIER message (if messages are enabled).

S8 Pause time generated by comma.
 0-255 Default=1 Units=Seconds. R/W.

S9 Carrier detect response time.
 0-255 Default=6 Units=Seconds/10 R/W.

The time a carrier must be present for the modem to accept it as carrier. This is different to S7, and is only used when communication has already been established and carrier has then been lost. This value defines how long the modem will wait, after re-detecting carrier before it continues with the communication.

S10 Carrier loss response time.
 0-255 Default=10 Units=Seconds/10 R/W.

The time between carrier disappearing and the modem hanging up. This allows for brief line dropouts. This value must be greater than the value in S9 or carrier will never be re-detected after dropping out. The maximum carrier dropout length is therefore

$$((S10)-(S9)) \times 100\text{mS.}$$

S11 Tone mark and space periods.
 50-255 Default=80 Units=Seconds/1000 R/W

This register holds the value of the touch tone digit period and the inter-digital delay, which are always the same value.

S12 Escape code guard time.
 0-255 Default=50 Units=Seconds/50 R/W.

The time period before and after the escape sequence for which no characters must be received. The function of the guard time is to minimise the possibility of the escape sequence occurring during a normal transmission.

S13 Reserved for internal use.

Bit-mapped. Read only.

Bit: Parameter: Command:

1 0-Basic result code set X
 1-Extended result code set

2 0-Parity disabled -
 1-Parity enabled

3 0-Odd parity -
 1-Even parity

4 0-7 data bits -
 1-8 data bits

5	0-Parity fixed 1-Parity active (moving)	-
6	0-No buffer overflow 1-Buffer overflow	-

S14 Status register.

Bit-mapped. Read only.

Bit:	Parameter:	Command:
1	0-Command echo off 1-Command echo on	E
2	0-Result codes sent 1-Result codes not sent	Q
3	0-Numeric result codes 1-ASCII result codes	V
5	0-Tone dialling 1-Pulse dialling	T/P
7,6	7 6 00 -Monitor off always. 0 1 -Monitor on until carrier detected. 1 0 -Monitor on always.	M

S15 Reserved for internal use.

Bit-mapped. Read only.

Bit:	Parameter:	Command:
6	0-Transmitter off 1-Transmitter on	C

S16 Reserved for internal use.

Bit-mapped. Default=0 Read only.

Bit:	Parameter:	Command:
6	0-Analog loopback with self-test disabled.	&T0
	1-Analog loopback with self-test enabled.	&T8

S17 Reserved for internal use.

Bit-mapped.

Read only.

This register contains the country definition in its lowest four bits.

3210	Country
0000	UK
0001	USA

S18 Test timer.

0-255

Default=0

Units=Seconds

R/W.

This timer controls the duration of loopback and self tests. If the default value of 0 is used, tests last for ten seconds. If any other value is used, tests will last for that number of seconds.

S19 Mode setting.

1,4,5,8,17,20,21,24

Default=5

Read Only.

The value in S19 is directly related to the value programmed by the ATB command. Initially, the value programmed is entered into S19. During the course of a call setup, the value may be changed automatically, by the modem, e.g. from originate to answer, or from V22bis 2400 bps to V22 bis 1200 bps.

4. RESULT CODES

BASIC SET

Numeric:	ASCII:	Meaning:
0	OK	Command line executed without errors.
1	CONNECT	Carrier detected.
2	RING	Ringling signal detected.
3	NO CARRIER	Carrier been lost or never detected.
4	ERROR	Incorrect/invalid command line entered.

EXTENDED SET

5	CONNECT 1200	Carrier detected at 1200 baud.
6	NO DIAL TONE	No dial tone detected within timeout.
10	CONNECT 2400	Carrier detected at 2400 baud.

Result code notes:

- i) Certain modes allow 'fallback' to a lower speed if attempting to connect to a modem which is set up for the lower speed. If this happens, the modem will return a connect message at the high speed rate indicating that data transmission should take place at the lower rate, e.g. In V22bis (2400 bps) mode, at connect time, there is a protocol to allow possible fallback to 1200 bps (normal V22 which is compatible with BELL 212A). This will be handled by the modem returning a CONNECT 1200 result code at 2400 baud. (The host software will be expecting a CONNECT 2400 code and will need to respond to CONNECT 1200 by changing its baud-rates, in and out, to the lower value). This would occur, for instance, if the modem was set up to communicate at V22bis (2400 bps) and dialled into a BELL 212A service.
- ii) There is no CONNECT 300 result in the extended set because the default CONNECT is returned if a 300 baud carrier is detected and the extended results are selected.
- iii) NO DIAL TONE (6) is only returned for those countries where blind dialling is forbidden.

Modes for Hayes modem.

5. The modes provided are as follows-

(i)	Bell 103	300	bps	Originate.	Full duplex.
(ii)	"	"	"	Answer.	" "
(iii)	Bell 212	1200	bps	Originate.	" "
(iv)	"	"	"	Answer.	" "
(v)	V22bis	1200	bps	Originate.	" "
(vi)	"	"	"	Answer.	" "
(vii)	V22bis	2400	bps	Originate.	" "
(viii)	"	"	"	Answer.	" "

Baud-rate and serial format determination from AT command.

The modem monitors the AT commands received from the host computer and makes various decisions based on the data it receives.

The first decision is based on the baud-rate of the incoming data. If the current communications mode is incompatible with the detected baud-rate, the mode will **AUTOMATICALLY** be changed to the default mode for that speed. Default modes are given below.

	300	1200	2400
Mode	Bell 103 Orig.	Bell 212A Orig.	V22 bis Orig.

The second decision is based on the serial format of the incoming data. It is not possible to determine all possible formats, but the following can be handled **IF THE AT PART OF THE COMMAND IS IN UPPER CASE** (lower case or mixed upper and lower case will considerably reduce the ability of the modem to determine the incoming format and are best not used where possible):

7 data bits, odd, even, mark, space or no parity.

8 data bits, odd, even, space or no parity.

[Notes-

- (i) 7-S will be detected as 8-N, but this will not cause any error.
- (ii) For 7-N data, if the AT part comes in as a continuous string, it will be decoded correctly. If the two characters are separated e.g. typed in by hand, the modem will decode 7-M.]

The modem always works with one stop bit, which means that it will always send one, but can receive one or two with no difficulty.

AMSTRAD SOFTWARE LICENCE AGREEMENT

IMPORTANT NOTICE

DO NOT OPEN THE PACKAGE CONTAINING THE DISCS UNTIL YOU HAVE READ THIS LICENCE AGREEMENT CAREFULLY.

Amstrad Plc provides this Licence to the purchasers of this the Software to which this Licence Agreement relates ("Licensee"). Licensee should carefully read this Licence Agreement prior to breaking the seal on or opening the Package containing the Software. By opening the package or breaking the seal, Licensee agrees to be bound by all the terms and conditions of this Agreement. If you do not find the terms and conditions of this Agreement acceptable, return the complete Package UNOPENED within ten (10) working days of purchase to the company which supplied it to you, and that company will refund or credit the full price of the Software to you.

DEFINITIONS

1. **"Machine"** means the single microcomputer, or PC on which you use the Software. Multiple CPU systems require additional licences.
2. **"Software"** means the set of programs, discs, tapes, documentation and related materials contained in and with the Package.
3. **"Amstrad"** means Amstrad Inc, 1915 Harrison Road, Longview Tx 75604.
4. **"Licensee"** means the purchaser if purchased for own use or, if purchased for company use, the purchaser, that company and its employees.
5. **"Package"** means the packaging in which the Software is contained.

LICENCE

Amstrad grants a non-exclusive Software Licence to Licensee to:

1. **Use the supplied Software on a single Machine**
 2. **Copy the Software into any machine-readable form for backup or modification purposes in support of Licensee's use of the Software on a single Machine. Any portion of the Software so copied will continue to be subject to the terms of this Licence. Licensee may make only one (1) copy of the Machine readable Software for such purposes. Copying of documentation and other printed materials is prohibited. Disassembly, reverse compilation and reverse engineering of the Software is prohibited.**
-

TERMS

1. Software supplied by Amstrad is copyrighted. Licensee agrees to uphold these copyrights.
2. This licence agreement enters into effect at the time you open or break the seal on the Package and is effective until terminated.
3. Licensee may terminate this licence Agreement at any time by destroying the Software together with all copies. The benefit derived by the Licensee from the Licence will terminate automatically if the terms of this Agreement are violated, and Amstrad may demand the return of the Software immediately. Any such termination shall be without prejudice to any accrued rights of the parties.
4. This Licence is personal to the Licensee and may not be assigned to any other person, persons or company.
5. The right to "rent", "sell" or otherwise transfer the Software in part or in whole is not granted to the Licensee.
6. Licensee may not use, copy, modify, or transfer the Software or documentation, or any copy, modification or merged portion, in whole or in part, except as expressly provided for in this Licence Agreement.
7. The Software cannot be transferred via any media other than that on which it is supplied. It cannot therefore be transferred via such media as telecommunication lines.
8. Licensee agrees to take all possible steps to protect the Software from unauthorised use, reproduction or distribution.

LIMITED WARRANTY

THE SOFTWARE IS PROVIDED ON AN "AS IS" BASIS. Except as is herein expressly stated, Amstrad make no representation and gives no warranty of any kind whatsoever either express or implied, including, but not limited to the implied warranties of merchantability and fitness for purpose (specific or general).

AMSTRAD DOES NOT WARRANT that the functions contained in the Software will meet Licensee's requirements or that the operation of the program will be uninterrupted or error free.

HOWEVER Amstrad warrants the discs, tapes or other media on which the program is embedded, to be free from defects in materials and workmanship under normal use for a period of ninety (90) days from the date of receipt of delivery to Licensee as evidenced by a dated copy of Licensee's proof of purchase receipt. In the event that the discs, tapes or media are found to be so defective within the said period, Amstrad will replace them free of charge, if they are returned postage prepaid.

Amstrad will also **AT ITS DISCRETION**, by modification or alteration, endeavour to remedy any program errors, attributable to Amstrad, which are provable and demonstrable, and reported to Amstrad in writing within the aforesaid period.

LIMITATIONS

Under no circumstances will the manufacturer, developer or supplier of the licensed Software be liable for any consequential, incidental, indirect, special or exemplary damages or other economic or physical loss, even if Amstrad has been advised of the possibility of such damages and loss.

Under no circumstances will liability exceed the purchase price of the Software.

INDEMNITY

The Licensee shall indemnify Amstrad in respect of any liability incurred by Amstrad as a result of a claim brought by the Licensee against Amstrad directly or against a third party, for negligence or breach of contract.

GOVERNING LAW

This Agreement shall be construed, interpreted and governed in accordance with the Laws of New York.

GENERAL

If any of the above provisions or portions thereof are invalid or unlawful under any applicable Law or Act of Parliament, they are to that extent only deemed omitted.

PPC ORGANIZER SOFTWARE LICENCE AGREEMENT

IMPORTANT NOTICE

DO NOT OPEN THE PACKAGE CONTAINING THE DISCS UNTIL YOU HAVE READ THIS LICENCE AGREEMENT CAREFULLY.

Amstrad Plc provides this Licence to the purchasers of this the Software to which this Licence Agreement relates ("Licensee"). Licensee should carefully read this Licence Agreement prior to breaking the seal on or opening the Package containing the Software. By opening the package or breaking the seal, Licensee agrees to be bound by all the terms and conditions of this Agreement. If you do not find the terms and conditions of this Agreement acceptable, return the complete Package UNOPENED within ten (10) working days of purchase to the company which supplied it to you, and that company will refund or credit the full price of the Software to you.

DEFINITIONS

1. **"Machine"** means the single microcomputer, or PC on which you use the Software. Multiple CPU systems require additional licences.
2. **"Software"** means the set of programs, discs, tapes, documentation and related materials contained in and with the Package.
3. **"Amstrad"** means Amstrad Inc, 1915 Harrison Road, Longview TX 75604.
4. **"Licensee"** means the purchaser if purchased for own use or, if purchased for company use, the purchaser, that company and its employees.
5. **"Package"** means the packaging in which the Software is contained.

LICENCE

Amstrad grants a non-exclusive Software Licence to Licensee to:

1. **Use the supplied Software on a single Machine**
 2. **Copy the Software into any machine-readable form for backup or modification purposes in support of Licensee's use of the Software on a single Machine. Any portion of the Software so copied will continue to be subject to the terms of this Licence. Licensee may make only one (1) copy of the Machine readable Software for such purposes. Copying of documentation and other printed materials is prohibited. Disassembly, reverse compilation and reverse engineering of the Software is prohibited.**
-

TERMS

1. Software supplied by Amstrad is copyrighted. Licensee agrees to uphold these copyrights.
2. This licence agreement enters into effect at the time you open or break the seal on the Package and is effective until terminated.
3. Licensee may terminate this licence Agreement at any time by destroying the Software together with all copies. The benefit derived by the Licensee from the Licence will terminate automatically if the terms of this Agreement are violated, and Amstrad may demand the return of the Software immediately. Any such termination shall be without prejudice to any accrued rights of the parties.
4. This Licence is personal to the Licensee and may not be assigned to any other person, persons or company.
5. The right to "rent", "sell" or otherwise transfer the Software in part or in whole is not granted to the Licensee.
6. Licensee may not use, copy, modify, or transfer the Software or documentation, or any copy, modification or merged portion, in whole or in part, except as expressly provided for in this Licence Agreement.
7. The Software cannot be transferred via any media other than that on which it is supplied. It cannot therefore be transferred via such media as telecommunication lines.
8. Licensee agrees to take all possible steps to protect the Software from unauthorised use, reproduction or distribution.

LIMITED WARRANTY

THE SOFTWARE IS PROVIDED ON AN "AS IS" BASIS. Except as is herein expressly stated, Amstrad make no representation and gives no warranty of any kind whatsoever either express or implied, including, but not limited to the implied warranties of merchantability and fitness for purpose (specific or general).

AMSTRAD DOES NOT WARRANT that the functions contained in the Software will meet Licensee's requirements or that the operation of the program will be uninterrupted or error free.

HOWEVER Amstrad warrants the discs, tapes or other media on which the program is embedded, to be free from defects in materials and workmanship under normal use for a period of ninety (90) days from the date of receipt of delivery to Licensee as evidenced by a dated copy of Licensee's proof of purchase receipt. In the event that the discs, tapes or media are found to be so defective within the said period, Amstrad will replace them free of charge, if they are returned postage prepaid.

Amstrad will also **AT ITS DISCRETION**, by modification or alteration, endeavour to remedy any program errors, attributable to Amstrad, which are provable and demonstrable, and reported to Amstrad in writing within the aforesaid period.

LIMITATIONS

Under no circumstances will the manufacturer, developer or supplier of the licensed Software be liable for any consequential, incidental, indirect, special or exemplary damages or other economic or physical loss, even if Amstrad has been advised of the possibility of such damages and loss.

Under no circumstances will liability exceed the purchase price of the Software.

INDEMNITY

The Licensee shall indemnify Amstrad in respect of any liability incurred by Amstrad as a result of a claim brought by the Licensee against Amstrad directly or against a third party, for negligence or breach of contract.

GOVERNING LAW

This Agreement shall be construed, interpreted and governed in accordance with the Laws of New York.

GENERAL

If any of the above provisions or portions thereof are invalid or unlawful under any applicable Law or Act of Parliament, they are to that extent only deemed omitted.

MIRROR II SOFTWARE LICENCE AGREEMENT

IMPORTANT NOTICE

DO NOT OPEN THE PACKAGE CONTAINING THE DISCS UNTIL YOU HAVE READ THIS LICENCE AGREEMENT CAREFULLY.

Amstrad Plc provides this Licence to the purchasers of this the Software to which this Licence Agreement relates ("Licensee"). Licensee should carefully read this Licence Agreement prior to breaking the seal on or opening the Package containing the Software. By opening the package or breaking the seal, Licensee agrees to be bound by all the terms and conditions of this Agreement. If you do not find the terms and conditions of this Agreement acceptable, return the complete Package UNOPENED within ten (10) working days of purchase to the company which supplied it to you, and that company will refund or credit the full price of the Software to you.

DEFINITIONS

1. "Machine" means the single microcomputer, or PC on which you use the Software. Multiple CPU systems require additional licences.
2. "Software" means the set of programs, discs, tapes, documentation and related materials contained in and with the Package.
3. "Amstrad" means Amstrad Inc, 1915 Harrison Road, Longview Tx 75604.
4. "Licensee" means the purchaser if purchased for own use or, if purchased for company use, the purchaser, that company and its employees.
5. "Package" means the packaging in which the Software is contained.

LICENCE

Amstrad grants a non-exclusive Software Licence to Licensee to:

1. Use the supplied Software on a single Machine
 2. Copy the Software into any machine-readable form for backup or modification purposes in support of Licensee's use of the Software on a single Machine. Any portion of the Software so copied will continue to be subject to the terms of this Licence. Licensee may make only one (1) copy of the Machine readable Software for such purposes. Copying of documentation and other printed materials is prohibited. Disassembly, reverse compilation and reverse engineering of the Software is prohibited.
-

TERMS

1. Software supplied by Amstrad is copyrighted. Licensee agrees to uphold these copyrights.
2. This licence agreement enters into effect at the time you open or break the seal on the Package and is effective until terminated.
3. Licensee may terminate this licence Agreement at any time by destroying the Software together with all copies. The benefit derived by the Licensee from the Licence will terminate automatically if the terms of this Agreement are violated, and Amstrad may demand the return of the Software immediately. Any such termination shall be without prejudice to any accrued rights of the parties.
4. This Licence is personal to the Licensee and may not be assigned to any other person, persons or company.
5. The right to "rent", "sell" or otherwise transfer the Software in part or in whole is not granted to the Licensee.
6. Licensee may not use, copy, modify, or transfer the Software or documentation, or any copy, modification or merged portion, in whole or in part, except as expressly provided for in this Licence Agreement.
7. The Software cannot be transferred via any media other than that on which it is supplied. It cannot therefore be transferred via such media as telecommunication lines.
8. Licensee agrees to take all possible steps to protect the Software from unauthorised use, reproduction or distribution.

LIMITED WARRANTY

THE SOFTWARE IS PROVIDED ON AN "AS IS" BASIS. Except as is herein expressly stated, Amstrad make no representation and gives no warranty of any kind whatsoever either express or implied, including, but not limited to the implied warranties of merchantability and fitness for purpose (specific or general).

AMSTRAD DOES NOT WARRANT that the functions contained in the Software will meet Licensee's requirements or that the operation of the program will be uninterrupted or error free.

HOWEVER Amstrad warrants the discs, tapes or other media on which the program is embedded, to be free from defects in materials and workmanship under normal use for a period of ninety (90) days from the date of receipt of delivery to Licensee as evidenced by a dated copy of Licensee's proof of purchase receipt. In the event that the discs, tapes or media are found to be so defective within the said period, Amstrad will replace them free of charge, if they are returned postage prepaid.

Amstrad will also **AT ITS DISCRETION**, by modification or alteration, endeavour to remedy any program errors, attributable to Amstrad, which are provable and demonstrable, and reported to Amstrad in writing within the aforesaid period.

LIMITATIONS

Under no circumstances will the manufacturer, developer or supplier of the licensed Software be liable for any consequential, incidental, indirect, special or exemplary damages or other economic or physical loss, even if Amstrad has been advised of the possibility of such damages and loss.

Under no circumstances will liability exceed the purchase price of the Software.

INDEMNITY

The Licensee shall indemnify Amstrad in respect of any liability incurred by Amstrad as a result of a claim brought by the Licensee against Amstrad directly or against a third party, for negligence or breach of contract.

GOVERNING LAW

This Agreement shall be construed, interpreted and governed in accordance with the Laws of New York.

GENERAL

If any of the above provisions or portions thereof are invalid or unlawful under any applicable Law or Act of Parliament, they are to that extent only deemed omitted.

MS DOS 3.3

Microsoft Corporation and AMSTRAD END USER PROGRAM LICENCE AGREEMENT

NOTICE TO USER - PLEASE READ THIS NOTICE CAREFULLY. DO NOT OPEN THE DISKETTE PACKAGE UNTIL YOU HAVE READ THIS LICENCE AGREEMENT.

OPENING THE DISKETTE PACKAGE INDICATES YOUR AGREEMENT TO BE BOUND BY THESE TERMS AND CONDITIONS.

AMSTRAD provides this program and licenses its use. You assume responsibility for the selection of the program to achieve your intended results, and for the installation, use and results obtained from the program.

LICENCE

You may:

- 1 Use the program on a single machine;**
- 2 Copy the program into any machine readable or printed form for backup or modification purposes in support of your use of the program on a single machine.** (Certain programs, however, may include mechanisms to limit or inhibit copying. They are marked "copy protected".);
- 3 Modify the program and/or merge it into another program for your use on the single machine.** (Any portion of this program merged into another program will continue to be subject to the terms and conditions of this Agreement); and,
- 4 Transfer the program and licence to another party if the other party agrees to accept the terms and conditions of this Agreement.** If you transfer the program, you must at the same time either transfer all copies, including the original, whether in printed or machine readable form to the same party, or destroy any copies not transferred; this includes all modifications and portions of the program contained or merged into other programs.

You must reproduce and include the copyright notice on any copy, modification or portion merged into another program.

YOU MAY NOT USE, COPY, MODIFY OR TRANSFER, THE PROGRAM OR ANY COPY, MODIFICATION OR MERGED PORTION IN WHOLE OR IN PART, EXCEPT AS EXPRESSLY PROVIDED FOR IN THIS LICENSE AGREEMENT.

IF YOU TRANSFER POSSESSION OF ANY COPY MODIFICATION OR MERGED PORTION OF THE PROGRAM TO ANOTHER PARTY YOUR LICENCE IS AUTOMATICALLY TERMINATED.

TERM

The licence is effective until terminated. You may terminate it at any other time by destroying the program together with all copies, modifications and merged portions in any form. It will also terminate upon conditions set forth elsewhere in this Agreement or if you fail to comply with any term or condition of this Agreement. You agree upon such termination to destroy the program together with all copies modifications and merged portions in any form.

LIMITED WARRANTY

THE PROGRAM IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU (AND NOT AMSTRAD) ASSUME THE ENTIRE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

AMSTRAD does not warrant that the functions contained in the program will meet your requirements or that the operation of the program will be uninterrupted or error free.

However, AMSTRAD warrants the diskette(s) on which the program is furnished, to be free from defects in materials and workmanship under normal use for a period of ninety (90) days from the date of delivery to you as evidenced by a copy of your receipt.

LIMITATIONS OF REMEDIES

AMSTRAD's entire liability and your exclusive remedy against AMSTRAD shall be:

- 1 The replacement of any diskette not meeting AMSTRAD's "Limited Warranty" and which is returned to AMSTRAD with a copy of your receipt.**
- 2 If AMSTRAD is unable to deliver a replacement diskette which is free of defects in materials or workmanship, you may terminate this Agreement by returning the program and your money will be refunded.**

IN NO EVENT WILL AMSTRAD BE LIABLE FOR ANY DAMAGES, INCLUDING ANY LOST PROFITS, LOST SAVINGS OR OTHER INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE SUCH PROGRAM EVEN IF AMSTRAD HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES, OR FOR ANY CLAIM BY ANY OTHER PARTY.

GENERAL

You may not sublicense, assign or transfer the licence or the program except as expressly provided in this Agreement. Any attempt otherwise to sublicense, assign or transfer any of the rights, duties, or obligations hereunder is void.

This Agreement shall be governed by and construed in accordance with the laws of the State of Washington.

Should you have any questions concerning this Agreement, you may contact AMSTRAD by writing to Amstrad Inc, 1915 Harrison Road, Longview TX 75604.

YOU ACKNOWLEDGE THAT YOU HAVE READ THIS AGREEMENT, UNDERSTAND IT AND AGREE TO BE BOUND BY ITS TERMS AND CONDITIONS. YOU FURTHER AGREE THAT IT IS THE COMPLETE AND EXCLUSIVE STATEMENT OF THE AGREEMENT BETWEEN YOU AND AMSTRAD WHICH SUPERSEDES ANY PROPOSAL OR PRIOR AGREEMENT, ORAL OR WRITTEN, AND ANY COMMUNICATIONS BETWEEN US RELATING TO THE SUBJECT MATTER OF THIS AGREEMENT.

THIS AGREEMENT DOES NOT AFFECT YOUR STATUTORY RIGHTS.

INDEX

- ABort 279 285
 - ACept 282 285
 - access protection 282
 - ALarm 279 285
 - AMSTRAD PPC
 - firmware 309
 - reference material 309
 - ROM 309
 - ROS (Resident Operating System) 309
 - setting up your PPC 7
 - software disks 301
 - ANSI screen codes 309
 - ANSI.SYS 310
 - ANsback 285
 - answer mode operation
 - general 255 256 257 281
 - Password maintenance Utility 282
 - remote commands 283
 - APPEND (DOS) 88 92
 - appending to a disk file (MIRROR) 259
 - APrefix 285
 - archive attribute 147 149
 - ASSIGN 80 93
 - ASk 279 281 285
 - ATention 249 251 255 256 259 263 274 285
 - attention key 249 251 255 256 259 263 274 285
 - ATTRIB 147 149
 - attributes 147 149
 - automatic SPeed settings 258
 - AUTOEXEC.BAT 120

 - BAckgrnd 272 285
 - MIRROR background operation 252 272
 - setting the background keys 272
 - back-up copies
 - AMSTRAD PPC disks 31
 - files 140 146
 - baud rate (see also SPeed) 255 257
 - BATch files 116 121
 - conditional commands 117 128
 - dummy parameters 119
 - ECHO 116 124
 - FOR 116 126
 - GOTO 116 128
 - IF 116 129
 - labelled points in a BATch file 128
 - multi-purpose BATch files 117
 - PAUSE 118 131
 - REM 118 132
 - repeated commands 117 126
 - screen messages 117 121 132
 - SHIFT 120 133 what to include 298
 - BATch processing 116 121
 - stopping a batch process 120
 - batteries 9
 - binary file (see also protocols)

 - BKsize 285
 - BLankex (blank lines) 263 285
 - bleep 13
 - BREAK 86 94
 - BReak 266
 - break signal 260 286
 - browsing
 - capture buffer 261
 - disk files 274
 - BUFFERS command 322
 - buffer size, capture 259
 - BYe 235 252 286

 - cable for direct data transfer 231
 - call progress monitoring (see CProgress)
 - CApture 249 251 258 281 286
 - command completion code 281
 - pausing 249 250 261
 - retro 260
 - terminating 249 269
 - capture buffer 260
 - browsing 261
 - EDiting 261 274
 - erasing 260
 - printing 262
 - size adjustment 259
 - WRiting it to disk 260
 - capturing data
 - to disk 259
 - to memory 259
 - CDir 283 286
 - characher echo 263
 - character filtering 260
 - character keys 20
 - character sets 305
 - internal codes 306 307
 - keystrokes 308
 - CHDIR 79 95
 - checksum 263
 - CHKDSK 187 189
 - cleaning your PPC 50
 - CLear 279
 - clearing the screen 83 97
 - clock
 - setting operating system clock 19
 - setting the date 20 204 206
 - setting the time 19 204 215
 - CLS 83 97
 - color, screen use 283 286
 - COM files 46
 - COM 1, COM 2 201 210 316
 - combining files, using COPY 140 152
 - COmmand 255 286
 - command completion register 281
 - command files 250
 - file type 273
-

- LOading 250 273 289
- modifying 273
- SAving 273 291
- command line
 - DOS 54 58
 - MIRROR 229 238 249 253
- command line editing
 - DOS 83
 - MIRROR 253
- command name 54 55
 - details of program location 82
- command summary
 - DOS 217
 - MIRROR 295
- command tail 54 58
- COMMAND.COM 186
- command usage 285
- communications links (via the serial port)
 - setting up, using MODE 201 209
- COMP 144 151
- comparing disks 188 191
- comparing files 144 151
- computing fundamentals 1 41
- Compuserve 241 244
- CONFIG.SYS 322
- CONNECT 242 351
- connect time (see on line timer)
- contrast control 14
- control characters
 - filtering out 260
 - translating 260
- conventions
 - DOS 69
 - MIRROR 247
- COPY 140 152
- copy-protected programs 296
- copying disks 33 186 193
- copying files 35
 - from input devices 140 153
 - to output devices 140 153
 - using copy 140 153
- CProgress (call progress monitoring) 258 286
- CROSTALK 235 238 239
- CRc 234 239 264 267 286
- CStatus 261 286
- CTTY 200 205
- current directory 75
 - changing the current directory 78 98
- currently logged MIRROR DRIVE (see DRive)
- cursor keys 20
- CWait 274 296

- DAta bits 258 286
- DATE 20 204 206
- date and time stamping 147
- DEbug 260 286
- default directory 4
- default drive 4 57
- default MIRROR disk (see DRive)

- dialling procedure 250
- DEL 143 156
- deleting files 4 156
- device names 316
- DIR (DOS) 75 98
- Dir(MIRROR) 256 283 287
- directories
 - creating new 137 166
 - current 78
 - default 4
 - names 47
 - structure 136
 - listing 75 98
 - removing 139 175
 - root 4
- disk buffers 322
- DISKCOMP 191
- DISKCOPY 193
 - disk drives 15 23
 - troubleshooting 327 339
- disk housekeeping 31 140
- display selector switches 10 317
- display window (see window)
- DNames 287
- DO 256 279 287
- documents 5
- DOS programs 27 29 30
- downloading (see capturing data) 251
- DPrefix 287
- DRive 283 287
- DSuffix 287
- DUpIex 258 287

- ECHO 117 124
- echoing to the printer 86
- EDit 261 274 287
- editing files 4 39 140 158
- editing, command line
 - DOS 83
 - MIRROR 253
- Editor (MIRROR's)
 - block commands 277
 - block marks 277
 - capture buffer 274 275
 - command formats 275
 - control characters 276
 - cursor positioning 275
 - delete commands 276
 - editing files 274
 - exiting the Editor 274
 - find string 276
 - insert commands 276 277
 - print function 278
 - replace string 276
 - save in place 275
 - saving data to disk 275
 - scrolling 275
- EDLIN 140 158
- EDLIN commands 141

- EMulate 232 272 287
- ENd 279 287
- entering commands 249
- environment parameters 82 109 119
- EPath 287
- ERASE (DOS) 143 156
- ERase (MIRROR) 283 287
- error messages
 - hardware 339
 - operating system 331
- ESCAPE key (see ATtention)
- EXE files 46
- external commands 57

- FC command 144 162
- file name templates 46
- file names 45
- file organisation 76
- file transfers 226 229
- files 2
 - attributes 147 149
 - combining files using COPY 141 153
 - comparing files 144 151
 - copying, using copy 140 153
 - date and time stamping (DOS media) 147
 - deleting, using MS-DOS 143 156
 - editing 39 140 158
 - erasing, using MS-DOS 143 156
 - illegal file names 45
 - listing 142 179
 - moving 140
 - number of open files 323
 - printing 142 read-only files 147 149
 - read-write files 147 149
 - renaming, using MS-DOS 172
 - size of 38 187 189
 - specifying a group of files 46
 - time and date stamping (DOS media) 147
 - valid filenames 45
 - write protecting 147 149 323
- FILES command 323
- filters 91 100
- Filter 256 260 288
- filter table 256
- filtering
 - blank lines 263
 - control characters 260 263
 - DEbug 260 286
 - tabs 263
- FIND 91 100
- FKeys 288
- FLow 260 263 288
- flow control, disabling 260 263
- function keys 255
 - displaying 255
 - general 255 282
 - maximum number of characters 282
 - setting 278 282
- FOR 117 126

- form statements 55
- FORMAT 185 195
- formatting disks 185 195
- free space 188
- function keys
 - DOS 20
 - MIRROR, displaying 255
 - general 255 282
 - HElP key, (F1)
 - maximum number of chars 282
 - setting 255 282
- fundamental calling steps 242

- garbage character elimination (see filtering)
- GKermit 270 288
- GO
 - command 244 250 273 274 279 281 284 288
 - RDial 257
- GOTO 117 128

- HAndshak 288
- hard copy transcript 259
- hardware
 - error messages 339
 - reference material 325
- Hayes 233 236 238 239
- Hayes protocol (see protocols)
- HElP 240 249 256 288
- hexadecimal numbers 305
- host 232

- IBM PC disks and the AMSTRAD PPC 27
- IF
 - DOS 117 129
 - MIRROR 279 281 288
- illegal filenames 45
- imaginary drives 80
- INfilter 256 260 288
- input and output devices
 - error messages 339
 - setting up, using MODE 202 209
- inserting a disk 25
- installing programs 296
- integrated text editor (MIRROR) 261
- intercharacter delay 264 286
- interfaces 2
- internal commands 62
- interrupting data transmission 260
 - restart 260
 - stop 260
- IO.SYS 186

- JOIN 81 104
- JUmp 279 281 288

- KErmit 239 268 288
 - commands 268 269
 - protocol 268 269
 - servers 268
 - settings 268 269
- keycodes 312
- KEYB 323
- keyboard 20 312
 - national keyboards 323
 - troubleshooting 329
- keys (MIRROR)
 - function (see function keys)
 - special 255

- LABEL (DOS) 204
- LAbel (MIRROR) 279 289
- LBreak 289
- LeaRn 250 274 289
- LFauto 260 289
- List 289
- LOad 250 256 273 289
- loading saved parameters 250 273
- log, transaction (see transaction log)
- logged disk (see DRive)
- lower case 249
- lower case conversion (see UOnly)
- LPT1, LPT2, LPT3, 201 316
- LWaiT 263 289

- macros (see function keys)
- making a call 250
- MESsage 289
- MInit 289
- MKDIR 137 166
- MODE (DOS) 202 209
- MOde (MIRROR) 250 257 266 281 283 289
- modem 233 244 340
- monitor 309 317
- MORE 91 106
- MS-DOS 41 51 53 54
- MS-DOS.SYS 185
- multi-tasking (see background operation)

- NAMe 281 289
- name conflicts 37
- nesting of Script files (see RTurn)
- NEWUSER 241
- new disks 23
- NUL 153
- NUmber 243 250 253 254 257 289

- on-line/off-line 235 250 253
- operating systems
 - error messages 331
 - reference material 325
 - version number 74 114
 - which to choose 296

- operating system-like commands 283
- OUtfiltr 263 290
- overtyping (command line editing) 253
- overwriting
 - disk files 259
 - during protocol transfers 265

- parallel printer port
 - pin-out 319
 - setting up using MODE 202 209
 - timing diagram 320
 - using the port 320
- parameter tables 254
- PArity 258 281 290
- Password 230 282
 - user security profiles 282
- Password Maintenance Utility 282
 - authorization levels 282
 - automatic Script file execution 282
 - executing the utility 282
- PATH (DOS) 87 107
- path (MIRROR, SEE CDir) 283
- paths
 - from the current directory 49 78
 - from the root directory 48 77
- PAUSE 118 131
- paused, capture mode 259
- PC-DOS 27
- peripherals 2
- PiCture 258 262 290
- pipinG 90
- placeholders 70 227
- PMode 290
- POrt 257 290
- port, serial 257
- ports 2 320
- post-processing output 90
- PRINT 142 168
- printer
 - setting up using MODE 202 209
 - troubleshooting 320
 - printer cable 321
- PRinter 258 262 283 290
- printinG
 - DOS 142 168
 - MIRROR
 - files 290
 - while capturing data 262 283
- PRN 316
- program filetypes 75
- program parameters 29
- program for the AMSTRAD PPC 27
- PROMPT 203 213
- protocols 250 264
 - Crosstalk 265
 - examples 271
 - Hayes 271
 - KERMIT 268

transfers 264
XMODEM
multi-file (Batch) 267
single-file 266
YMODEM 270
PWord 282 290

QUit 252 230 290

RBatch 269 271 290
RDial 257 290
read-only files 147 149
read-write files 147 149
redialing 257
redirecting
disk searches 81 104
piping 90
printer output 201 209
standard input 89
standard output 90
reference material 325
register (see command completion register)
REM 118 132
remote commands 283
removing a directory 139 175
RENAME 144 172
REPLACE 146 173
REply 280 290
reserved filenames 45
resetting your PPC 44
restart transmission character 263
retro-capture 260
RHayes 270 290
RKermit 270 291
RMDIR 139 175
RPED 39
RQuesT 267 291
RTurn 280 291
RUn 291
running programs
BATch files 116 121
DOS programs under MS-DOS 28 73 87
stopping a program 94 120
troubleshooting 328
turnkey programs 29
types of program 73
using MS-DOS 82
RWind 280 291
RXmodem 266 291
RYmodem 270 291

SAve 250 273 291
saving captured data 260
saving changes in parameters 250 273
SBreak 280 291
screen (DOS)
adjusting display 202 209
dumps 90 168

messages 117 121 132
SCreen 280 283 291
screen capture
Picture 262 290
SNAPSHOT 262 292
Script files 235 238 239
creating 279
executing (see also DO) 279
file type 279
nesting 279
sample 280
summary of commands 279 280
transaction log 281
SCreen selection 280 283
search paths
data files 88 92
program files 107 187
searching the capture buffer 261 276
security 282
SEnd 262 291
sending text files 256 263
serial port 257 318 320
serial printer, setting up, using MODE 201 209
SET 82 109
setting up your PPC 10
SHIFT 120 133
single drive systems, running 2 disk programs 81
size of files, displayed 38
size option of CAPture 259
SKip 280 291
Smartcom 270
SNAPSHOT 278 262 292
SORT 91 111
special keys 255
SPeed (see also baud rate) 252 255 257 292
startup disk, choosing 296
startup procedure 14 41 120
status indicator (on-line/off-line/local) 250 253
Status Screen 247 253 256
STop bits 258 292
stop transmission character 263
stopping
a BATch process 94
a call (see BYe) 252
a command in progress
(see also attention key, ESCape key)
a DOS program 94 120
a MIRROR program (see QUIT)
Script files (see attention key, ESCape key)
a session (see QUIT) 252
transmission of a file 265
storage space on disk, displayed 187 189
string searching (see also Editor) 261
stripping away control characters
INfilter 256 260 288
OUTfiltr (line-feeds) 263 290
SUBST 80 112
suspending data transmission (see flow control)
SWitch 255 284 292

- switch key 255
- switching off 43
- switching on 30
- syntax statements 55
- SYS 186 197
- system disks 197
- system prompt 55
- special prompt 203 213

- TAbex 263 292
- tabs, expanding (see TAbex)
- terminal mode 232 250
- Terminal Screen 251
- terminating (see also stopping) 252 260 265
- text editor (DOS) 39 140 158
- text editor (MIRROR)(See Editor)
- TIME 204 215
- Time 292
- time and date stamping 147
- TLog 281 292
- toggle commands 249
- transaction log 281
- transcript feature (see PPrinter)
- transmission modes 233
- transmission rate, changing (see baud rate, SPeed) 255 263
- transmitting files (see protocols; see also SEnd)
- TREE 137 177
- trouble shooting
 - DOS 327
 - MIRROR
 - calling or logging on to another computer 251
 - SEnding to slower systems 264
- TUrnamd 292
- turnkey programs 29 298
- TYPE 142 179
- TYpe 283 292
 - capture buffer 256 261
 - disk files (text only) 256 261 283
 - interrupting 261

- UCOnly 263 292
- uploading (see also sending text files)
- upper case 249
- VER 74 114
- VERIFY 83 115
- verification protocol (see protocols)
- verifying data on disks 83 188 189
- VERsion 292Video 293
- VOL 181
- volume control 14

- WAt 280 293
- WHen 280 293
- wildcards 46 66
- window 255 257
 - CLearing 256
 - general 253 256
- WRite 260 281 293

- write-protecting floppy disks 24
- write protection 147 149
- XBatch 267 293
- XCOPY 140 182
- XDos 293
- XHayes 270 293

- XXermit 268 293
- XMit 265 266 293
- XModem file transfers
 - checksum 267 268
 - CRC algorithm 267 268
 - examples 267 268
 - multi-file 267 268
 - single file 266
- XON/XOFF handshaking 263
- XXmodem 266 293
- XYmodem 270 293
- YMODEM 239 270 293

Other doc available

PPC Tech Ref Manual (AMSTRAD)
ref p 322 & p 326
for CONFIG.SYS info

VT100 em switches

regular: /Wrap on off

VT100BC.mex / Cursor = pN / oH }

alt-c toggle

