



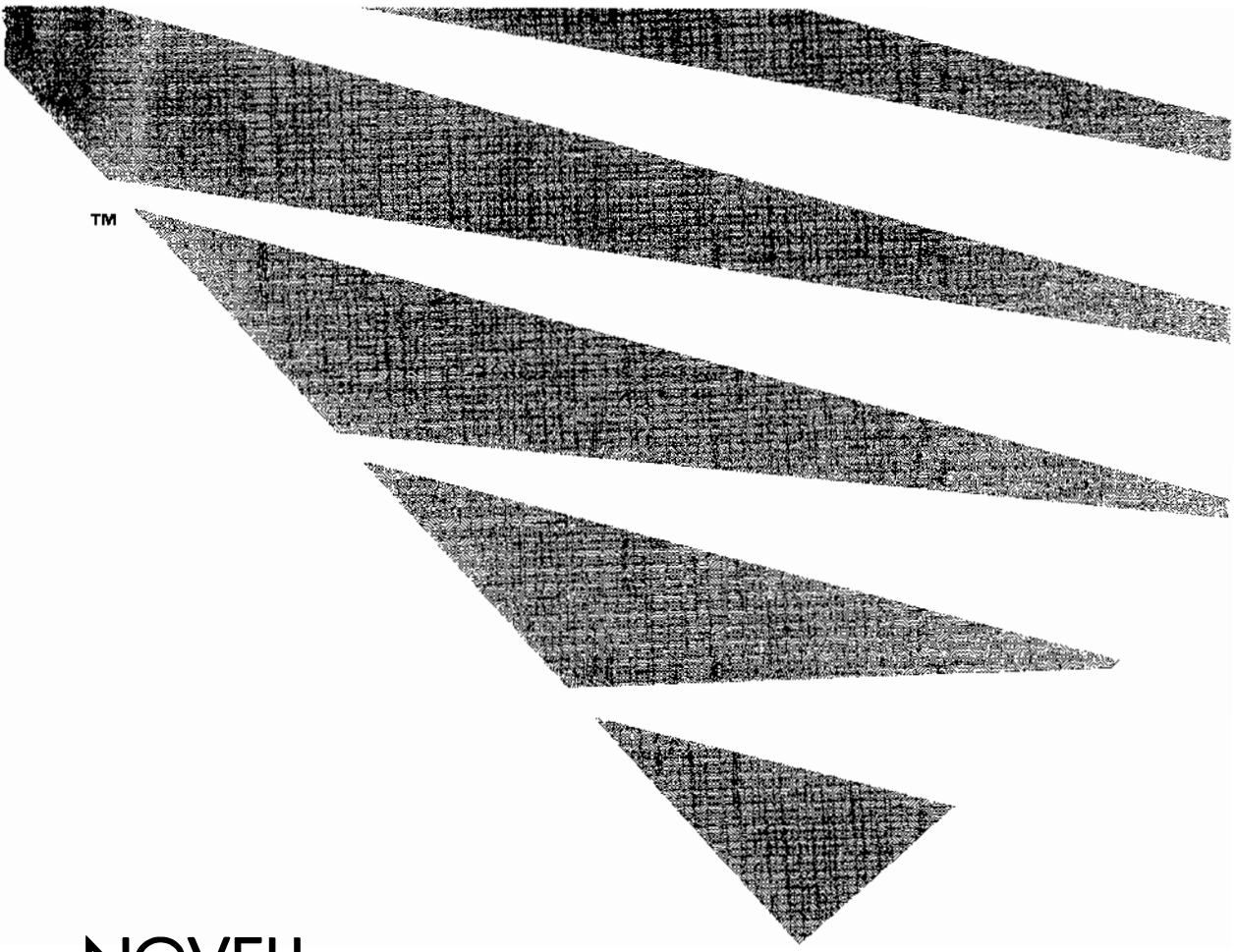
TM

NOVELL®

DR Multiuser DOS v.5.1

User's Guide

DR Multiuser DOS™



TM

NOVELL®

DR Multiuser DOS v.5.1

User's Guide

DR Multiuser DOS™

DISCLAIMER

DIGITAL RESEARCH INC. PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE. Further, Digital Research Inc. reserves the right to revise this publication and to make changes from time to time to the content hereof without obligation of Digital Research Inc. to notify any person of such revision or changes. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This publication could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. Digital Research Inc. may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time.

NOTICE TO USER

This manual should not be construed as any representation or warranty with respect to the software named herein. Occasionally changes or variations exist in the software that are not reflected in the manual. Generally, if such changes or variations are known to exist and to affect the product significantly, a release note or README file accompanies the manual and the distribution disks. In that event, be sure to read the release note or README file before using the product.

TRADEMARKS

Digital Research, its logo, and Digital Research Multiuser DOS are trademarks or registered trademarks of Digital Research, Inc..

Intel, 80386, 80386SX and 80486 are registered trademarks of Intel Corporation.

IBM and Personal System/2 are registered trademarks of International Business Machines Corporation.

Microsoft is a registered trademark, and MS-Windows a trademark, of Microsoft Corporation.

Paradox, Quattro Pro and Sidekick are registered trademarks of Borland International, Inc..

Lotus, 1-2-3 and Manuscript are registered trademarks of Lotus Development Corporation.

DataEase is a registered trademark of DataEase International, Inc..

Q&A is a registered trademark of Symantec Corporation.

WordPerfect and LetterPerfect are registered trademarks of WordPerfect Corporation.

dBASE IV and Framework are registered trademarks of Ashton-Tate Corporation.

Supercalc is a registered trademark of Computer Associates International, Inc..

Novell is a registered trademark of Novell, Inc..

VINES is a registered trademark of Banyan Systems, Inc..

AutoCAD is a registered trademark of Autodesk, Inc..

Wyse is a registered trademark of Wyse Technology.

ARNET and Smartport are registered trademarks of ARNET Corporation.

All other trademarks are the property of their respective holders and are hereby acknowledged.

COPYRIGHT

Copyright © 1976, 1982, 1988, 1990 and 1991 Digital Research Inc. All rights reserved. No part of this publication may be reproduced, or distributed without the prior written permission of Digital Research Inc., 70 Garden Court, P. O. Box DRI, Monterey, California 93942.

First edition: January 1991

Preface

THIS manual describes Digital Research Multiuser DOS, a multitasking, multiuser DOS operating system designed to run on Intel 80386, 80386SX and 80486® microprocessors.

It is intended for anybody that wishes to use a Multiuser DOS system. It also contains information specific to people who need to install and configure a system.

Finding your way around

The first part of this manual, “Installing,” is aimed at experienced users of personal computer systems who are responsible for setting up a working Multiuser DOS environment. The third part, “Configuring,” is similarly pitched; it contains information on how to fine tune an existing system.

If you are new to multitasking, multiuser operating systems, or indeed new to personal computers, you are unlikely to be charged with installing or maintaining your system. As such, you need only read Part 2 of this manual, “Using.” You should read the first five chapters of this part in order of content. They are written with the new user in mind, and build upon each other. The remaining chapters contain more detailed information on specific utilities offered by the operating system, together with a reference chapter covering all the available commands.

Below is a summary of the individual chapters and appendices.

Structure of the manual

Part 1 Installing

Chapter 1 Doing an installation

Describes the installation procedure and essential next steps. Includes information on configuring a system; this may be skipped and returned to at a later date.

Chapter 2 Setting up a secure system

Describes how to set up the operating system’s optional security features.

Chapter 3 Adding a PC as a terminal

Describes PCTERM, the terminal emulator program that lets PCs run as terminals in a multiuser environment.

Part 2 Using

Chapter 4 What is Multiuser DOS?

Gives an introduction to the main features of the operating system.

Chapter 5 Getting started

Describes the first practical steps to take to get acquainted with the operating system.

Chapter 6 Working with commands

Summarizes the operating system commands and ways of invoking them.

Chapter 7 Managing your information

Shows how to organize data into files and directories for ease of access, how to name files and directories and how to access them using paths. It also introduces some commands used to create and work with files and directories.

Chapter 8 Protecting your information

Describes the various ways of protecting your data from unauthorized access or accidental loss.

Chapter 9 The EDITOR

Describes the text editor EDITOR built in to the operating system.

Chapter 10 Batch processing

Explains how to create and work with batch files, which enable you to store frequently-used sets of commands in one file.

Chapter 11 Commands

Contains a description of all the operating system commands with their full command structure and function, along with examples of usage.

Chapter 12 Redirecting input and output

Shows how keyboard input can be sent directly to a printer or file, instead of to the screen, and output from one command used as input to another.

Chapter 13 Printing

Describes the various commands available for managing print operations.

Chapter 14 The SID debugging tool

Describes the SID program debugger.

Part 3 Configuring

Chapter 15 Running applications

Gives advice on how to maximize performance for individual applications, and gives worked examples of batch files that automate application calling.

Chapter 16 Customizing your system

Gives information on how to modify various configuration settings to optimize operating system performance. To be read in conjunction with Chapter 1.

Appendix A Error messages

Contains an explanation of error and system messages generated by the operating system.

Appendix B Connecting a serial terminal

Illustrates the correct pin settings for serial terminal connections.

Appendix C Adding a multiport board

Gives valid switch settings for various multiport boards, together with advice on how to use the ARNET device driver.

Appendix D Memory management

Summarizes the different types of memory used by the operating system.

Appendix E Using a setup input file

Shows how to reconfigure a system using a setup input file.

Appendix F Escape code sequences

Summarizes ANSI and VT52 escape sequences.

Appendix G Sample menuing batch file

Gives a sample batch file that sets up a simple menuing system for commonly-used utilities and applications.

Appendix H Glossary

Contains descriptions of technical terms found within the manual.

Assumptions and conventions

What you type in

What you should type in at your keyboard is shown in color, for example:

DISKCOPY a: a:

Examples are usually preceded by the system prompt, as follows:

A>DIR c: /S

Note that command names are shown in uppercase, as are command switches such as /S. This is done for the sake of clarity; in real life keyboard input appears on the screen in lowercase unless you use the **Shift** or **Caps Lock** key. The operating system ignores case when interpreting keyboard input.

Screen messages

Messages that appear on the screen are shown like this:

```
Insert diskette to format in drive A:  
and strike a key when ready
```

File entries

Sample entries in files, such as an extract from the **autoexec.bat** file, are shown as follows:

```
keyb  
date  
time
```

Special keys

The Enter key

The key to use to make the computer carry out an instruction is the **Enter** key, sometimes called the **Return** key or **Carriage Return** key. It often has the symbol ↵ on the keytop. In this manual it is always referred to as the **Enter** key.

Every command that you type at the keyboard needs to be followed by the **Enter** key. This is assumed in the main part of this manual. Sometimes when commands are being carried out, you may be prompted to respond to a question by typing a single letter, Y for YES or N for NO. In these cases you do not have to press **Enter**.

The Ctrl and Alt keys

The **Ctrl** (control) and **Alt** (alternative) keys are used to alter the meaning of another key. If you see **Ctrl+A**, you hold down the **Ctrl** key while typing **A**. Use the **Alt** key in the same way.

If you see **Ctrl+Alt+Del**, hold down **Ctrl** and **Alt**, and then press the **Del** key. This restarts ("reboots") the system.

Contents

Part 1 Installing

1 Doing an installation

1.1 Introduction	1-2
INSTALL and SETUP	1-2
1.2 Controlling the installation	1-3
Getting online help	1-3
Moving between screens	1-3
Setting values	1-3
Aborting installation	1-4
1.3 Starting INSTALL	1-4
1.4 Configuring the system	1-6
Main console and keyboard configuration	1-6
System directory	1-8
Memory configuration	1-8
System options and parameters	1-11
Serial port configuration	1-14
1.5 Installing system security	1-17
1.6 Copying the system	1-18
1.7 Rebooting	1-19
1.8 Logging in as Superuser	1-19
1.9 Copying your master diskettes	1-20

Twin-diskette-drive systems	1-21
1.10 Other tasks to perform	1-21

2 Setting up a secure system

2.1 User accounts	2-2
Access permissions	2-2
2.2 Adding users	2-3
2.3 Maintaining user accounts	2-5
2.4 Making the operating system partition secure	2-5
2.5 Making special files secure	2-6
2.6 Making commands secure	2-7
2.7 Words of advice	2-8

3 Adding a PC as a terminal

3.1 Introduction	3-2
PCTERM requirements	3-2
3.2 Running PCTERM	3-3
3.3 Setting terminal parameters	3-4

Part 2 Using

4 What is Multiuser DOS?

4.1 Multiuser support	4-2
Adding terminals	4-2
Resource sharing	4-2
Networking-in-a-box	4-3
Expanding the system	4-3
4.2 Multitasking	4-3
DOS sessions	4-4
4.3 Performance enhancement	4-4

4.4 Memory management	4-5
4.5 Multiuser DOS utilities	4-5

5 Getting started

5.1 Basic concepts	5-2
Disks	5-2
Drives	5-4
Files and directories	5-4
Memory	5-5
5.2 Using disks	5-6
Preparing disks for use	5-6
Write-protecting diskettes	5-6
The do's and don'ts of handling diskettes	5-8
5.3 Logging in	5-9
5.4 Getting started at the command line	5-10
The system prompt	5-10
The command line	5-10
Changing between drives	5-10
Typing instructions	5-11
Some simple commands	5-12
Getting help on commands	5-12
Running applications	5-13
5.5 Switching DOS sessions	5-13

6 Working with commands

6.1 Command types	6-2
Internal commands	6-2
External commands	6-2
Batch processing subcommands	6-4
Cconfig.sys commands	6-4

6.2 Entering commands	6-5
Command syntax	6-5
Specifying drives	6-6
6.3 Displaying data a screenful at a time	6-6
6.4 Stopping commands	6-6

7 Managing your information

7.1 Storing data in files	7-2
Naming files	7-2
Filenames you should not use	7-3
Handling groups of files (using wildcards)	7-4
Handling groups of files (using filelists)	7-5
7.2 Arranging files in directories	7-6
The current directory	7-7
Special directory symbols	7-8
Your home directory	7-8
7.3 Specifying paths	7-9
The PATH command	7-10
PATH and the autoexec.bat file	7-11
7.4 Working with directories	7-12
Listing directories (TREE)	7-12
Making a directory (MKDIR)	7-12
Displaying and changing directories (CHDIR)	7-13
Removing a directory (RMDIR)	7-13
7.5 General actions on files	7-14
Copying files (COPY)	7-14
Deleting files	7-15
Displaying file contents (TYPE)	7-15
Renaming files (RENAME)	7-16
Printing files	7-16

7.6 Keeping track of where you are	7-16
------------------------------------	------

8 Protecting your information

8.1 Changing your user password	8-2
8.2 Changing access permissions	8-3
Using XATTRIB	8-5
Using XATTRIB /F	8-6
8.3 The LOCK command	8-7
8.4 Password-protecting files and directories	8-8
Setting passwords	8-9
Issuing passwords	8-9
Password-protecting directories	8-9
Global system passwords	8-10
Removing passwords	8-10
8.5 Setting file attributes	8-12
8.6 Making backup copies of disks	8-13
Backing up your hard disk	8-13
Restoring backup files	8-16
Copying diskettes	8-17

9 The EDITOR

9.1 Starting EDITOR	9-2
9.2 Quitting EDITOR	9-2
9.3 Entering text	9-3
Using the Tab key	9-4
Moving text down a line	9-4
Entering control characters	9-4
Getting online help	9-4
9.4 Moving the cursor	9-5
To move the cursor one character	9-5

To move the cursor one word	9-5
To move the cursor one page	9-5
To move the cursor to the start or end of a line	9-6
To move the cursor to the start or end of a file	9-6
The cursor-control diamond	9-6
9.5 Inserting and overwriting text	9-7
Inserting a file at the cursor	9-7
9.6 Deleting text	9-7
Deleting a character	9-8
Deleting a word	9-8
Deleting a line	9-9
9.7 Working with blocks of text	9-10
Marking a block of text	9-10
Moving a block of marked text	9-10
Deleting a block of marked text	9-11
Copying a block of marked text	9-11
Storing a block of marked text in a file	9-11
9.8 Saving your work	9-11
9.9 Automatic backup files	9-12
9.10 Technical details	9-13
9.11 Command summary	9-14

10 Batch processing

10.1 Using batch files	10-2
10.2 Creating batch files	10-3
10.3 Running batch files	10-3
Stopping a batch file	10-4
10.4 Making general purpose batch files	10-4
Environment variables	10-6
10.5 Batch file subcommands	10-6

Command conventions	10-8
10.6 The autoexec.bat file	10-25
11 Commands	
11.1 Command summary	11-2
11.2 Command syntax explained	11-7
11.3 Editing commands at the command line	11-9
Command line editing keys	11-10
Control key combinations	11-11
11.4 Command reference	11-11
12 Redirecting input and output	
12.1 Using redirection	12-2
Redirecting output	12-2
Appending output	12-2
Redirecting input	12-2
Redirecting to or from a device	12-3
12.2 Using pipes and filters	12-4
Pipes	12-4
Filter commands	12-4
Combining pipes and redirection	12-6
13 Printing	
13.1 Using PRINTMAP	13-2
13.2 Running the CSPOOL print spooler	13-4
Installing CSPOOL	13-5
CSPOOL command options	13-6
Configuring the print spooler	13-9
Configuring DOS applications for the print spooler	13-14
Hints on using CSPOOL	13-15

13.3 Using MODE	13-16
Setting up a printer	13-16
Setting up a serial port	13-17
Assigning a printer	13-19
13.4 Using PRINT	13-20

14 The SID debugging tool

14.1 SID basics	14-2
Starting SID	14-2
Quitting SID	14-2
SID command conventions	14-2
Specifying a 20-bit address	14-3
Hexadecimal numbers	14-3
Decimal numbers	14-3
Character values	14-3
Operators in expressions	14-4
14.2 SID commands	14-5
14.3 Default segment values	14-35
14.4 Assembly language syntax (A and L commands)	14-38

Part 3 Configuring

15 Running applications

15.1 Overview	15-2
15.2 System-wide settings	15-3
Memory allocation	15-3
LIM memory allocation	15-3
Idle detection	15-4
Using printers	15-4
The command interpreter	15-5

15.3 DOS session settings	15-6
The MEMSIZE setting	15-6
The LIMSIZE setting	15-6
The BANK setting	15-6
The IDLE setting	15-7
The SUSPEND setting	15-7
The NETDRIVE setting	15-7
15.4 Application settings	15-8
Using PIFED	15-8
Setting up applications for terminals	15-13
Using a mouse	15-14
Configuring applications for printing	15-15
Direct memory access (DMA)	15-16
Updating the cconfig.sys file	15-17
Terminate-and-stay-resident (TSR) applications	15-17
Running multiple copies	15-18
Copy-protected applications	15-20
15.5 Worked examples	15-21
Preparing the system	15-21
Paradox v3.5	15-23
Microsoft Word v5.0	15-25
Lotus 123 Release 2.2 Server Edition	15-26
DataEase v4.2 LAN	15-28
Quattro Pro v2.0	15-29
Lotus Manuscript v2.1 LAN	15-31
WordPerfect v5.1	15-32
LetterPerfect v1.0	15-33
Q&A v3.0 LAN pack	15-34
SuperCalc 5	15-35

Framework III v1.1 15-36

dBASE IV v1.1 15-38

16 Customizing your system

16.1 Using SETUP	16-2
SETUP and system configuration files	16-3
SETUP with an input file	16-3
16.2 Cconfig.sys commands	16-4
16.3 Installable device drivers	16-29
16.4 The autoexec.bat file	16-36
Station and DOS session numbers as parameters	16-37
Starting applications from autoexec.bat	16-38
16.5 START files	16-39
16.6 Sharing resources	16-39
Printing	16-39
Sharing other resources	16-41

Appendices

A Error messages and troubleshooting

A.1 Error messages	A-2
A.2 Troubleshooting	A-44
Troubleshooting checklist	A-44
Diskette handling	A-45
Corrupted files	A-46
When absolutely nothing happens	A-46
Unresolvable errors	A-46

B Connecting a serial terminal

B.1 Null modem connections	B-2
----------------------------	-----

B.2 Possible configurations	B-3
Minimum configuration	B-3
Recommended configuration	B-3
 C Adding a multiport board	
C.1 Switch settings	C-2
C.2 The ARNET device driver	C-5
Installation	C-5
Handshaking	C-6
Baud rates	C-6
The /CACHE_FIX option	C-6
 D Memory management	
D.1 What is memory?	D-2
D.2 Types of memory	D-2
Conventional memory	D-2
High memory	D-4
Upper memory	D-5
Extended memory	D-6
LIM (expanded) memory	D-8
Non-banked memory	D-10
Checking on your computer's memory usage	D-11
Commands that control memory usage	D-11
 E Using a setup input file	
E.1 The setup input file	E-2
E.2 Input file values	E-2
SETMEM statements	E-4
SETMISC statements	E-5
SETMULTI statements	E-6

SETPORT statements	E-7
SETTERM statements	E-8

F Escape code sequences

F.1 ANSI escape sequences	F-2
F.2 VT52 Escape sequences	F-6

G Sample menuing batch file

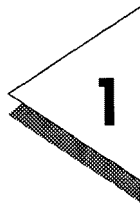
H Glossary

Index

Part 1

Installing

Doing an installation



THIS chapter explains how to install the operating system on your hard disk.

Read the whole chapter, and follow the instructions it contains, to perform an initial installation, or to reinstall the system. To fine tune an existing system, you should first read Chapter 16, “Customizing your system,” and then refer back to Section 1.4, “Configuring the system.”

1.1 Introduction

The INSTALL program

Doing an installation basically means copying the distribution diskettes, in the order in which they are numbered, onto the hard disk in your computer. It may also involve some setting of parameters, to map the operating system files to your particular hardware configuration.

A program called INSTALL is provided on the first diskette to take you through the installation procedure. INSTALL is menu-driven, easy to follow and has full context-sensitive help available at all stages.

INSTALL is designed to accommodate customers and system administrators who are new to multitasking, multiuser operating systems. It offers default installation values for each field. These set up a system on which almost all terminals and applications will run satisfactorily. Experienced installers are given the opportunity to set up a customized system straight away.

You can install the operating system on a new system with no files on the hard disk. You can also install the operating system over an existing version, but all the configuration files are overwritten and the system must be completely reconfigured.

You can also install this operating system on a system that already has a single-user DOS operating system installed. Note that if both operating systems are to be run on the same disk, the external DOS commands must be in a subdirectory, not the root directory of the disk.

INSTALL and SETUP

Once you have a system up and running, you can always modify your configuration settings, whether or not you have accepted the default values provided by INSTALL. To do this you run the SETUP program. SETUP offers the same functionality as INSTALL, but allows you to modify individual values without having to do a complete reinstall.

This chapter takes you through the installation procedure step by step, and concludes with some information on what to do on completing the installation. Section 1.4, "Configuring the system," describes the steps you can skip if you elect to install a

standard configuration. You can return to this section at a later date, when running **SETUP**.

1.2 Controlling the installation

To help you control the operation of the **INSTALL** program, you have at your disposal a range of keyboard functions. The exact combination of available functions may vary from one operation to the next. Functions currently available are displayed on the bottom line of the screen. This section summarizes the principal keyboard functions.

Getting online help

The **INSTALL** program provides context-sensitive help. To get help, move to the option or field for which you want help and press the **F1** key.

Moving between screens

To confirm any values displayed, and move to the next screen in the install operation, press the **Enter** key. If a range of options is available on the screen, pressing **Enter** always serves to confirm the highlighted option.

You can go back over the screens (and modify any setting you wish) by pressing the **Esc** key until you reach the screen you want.

Setting values

Each screen has at least one option to select. There may also be one or more fields that you can edit. (You can distinguish fields from options by their lack of the distinctive box to the left of them.)

An option can only be selected when it is highlighted. A field can only be edited when the cursor is positioned on it.

To move between options or fields, use the **Up** and **Down** arrow keys or the **Tab** key. As you move, the line at the bottom of the

screen may be updated to show which keys you can currently use.

Once you have the cursor in the field of your choice, you can edit its contents. To delete characters, use the **Del** or **Backspace** keys.

For help information on the keys to use to edit a field, position the cursor on the field and press **Alt+F1**.

The setting of a field that has two choices, for example YES/NO, can be selected using the **Spacebar**. If the default setting is YES, pressing the **Spacebar** once displays NO. Pressing it again displays YES and so on. This is called toggling.

In some fields there are several fixed choices. You can press the **Spacebar** to display a window showing the possible settings. Use the **Up** and **Down** arrow keys to highlight the value you require, then press **Enter** to select it. The window is removed and the value you selected is displayed in the field.

In both types of toggle field you can type in the value, rather than display and select it. You only need to type enough to make your choice clear; for example, if you typed OF in a ON/OFF field, OFF would be displayed. (Mistyping, for example OG, would clear the field for you to start again. Mistyping in a field with several settings displays the window for you to make a selection using the arrow keys.)

Aborting installation

You can quit the INSTALL program and discard any changes made by pressing the **F10** key. You will see a screen asking you if you really want to leave the program. Select the Exit to operating system option, then press the **Enter** key.

1.3 Starting INSTALL

You must install the operating system on the bootable hard disk partition of the main console (normally C:). (The operating system can only be installed on systems with a hard disk.) Follow these steps:

- 1** Insert the diskette identified as the first in the series into diskette drive A:. Close the drive door.
- 2** Switch on the computer or, if it is already on, press the **Ctrl+Alt+Del** keys at the same time to reboot your computer.
- 3** Wait for the first introductory screen of the **INSTALL** program to appear. Read its text, then press **Enter** to continue.
- 4** Partition the hard disk.

The **INSTALL** program checks to see whether your hard disk has been partitioned. If not, you need to partition your hard disk forthwith, using a program called **FDISK**. **INSTALL** automatically starts **FDISK** for you at this point if it determines that you require it.

WARNING **FDISK** should be used with care when preparing a hard disk. It is normally only used to prepare new disks.

For detailed instructions on how to use **FDISK**, refer to Chapter 11, “Commands.” When **FDISK** has finished, reboot your computer by pressing **Ctrl+Alt+Del** (ensure your first installation diskette is in diskette drive A:), then start again from step 3.

- 5** Install the default system or continue with system configuration.

You are now asked if you want to skip the configuration sequence and install a default system, or build a customized configuration.

If you elect to install the default system, you can skip the next section and rejoin the chapter at Section 1.5, “Installing system security.”

If you choose to continue with system configuration, **INSTALL** now takes you through a series of configuration screens. You can view the default values, and change them as necessary. Note that these are the same screens you are presented with if you run **SETUP** at a later date.

1.4 Configuring the system

In INSTALL (and SETUP) a series of screens are displayed, prompting you to supply new configuration values. This section goes through each group of screens, but does not describe each field in every screen in detail. For the extensive on-line help, simply press **F1**. You may not see all the screens in each group, as some screens are only displayed if you select a particular option at the beginning of the sequence.

Main console and keyboard configuration

This first screen is for your information only, and introduces the subject of national language versions of console and keyboard.

National versions

When running INSTALL, it is important to establish your national language at this stage, so that the driver for it can be loaded and the main console set to use that language before you complete the rest of the configuration screens. (If you are running SETUP and change the national language, the new driver is not loaded until you reboot the operating system.)

Press **Enter** to move to the next screen. You are now shown your default settings for country and keyboard.

Change country and/or keyboard

The country setting should indicate the principal national version of the operating system you wish to run. The keyboard setting should indicate the principal national language support you require for the keyboards in your configuration.

Normally you will wish the two settings to be the same. Note that it will always be possible on any terminal in your configuration to switch between the national keyboard support you set here and US English.

To change country or keyboard settings, select the appropriate option, then toggle through the available values until you have highlighted the one of your choice. Then press **Enter**.

When you are happy with country and keyboard settings, press **Enter** to confirm the current selections.

Main console configuration

The next screens ask you to configure the main console, the keyboard and screen, that is, on which you are currently working. (Other terminals are configured separately later in the program.)

You are presented with default settings and valid values for the following topics:

☐ Number of DOS sessions

This is the number of applications you can run at the same time. Each application runs in a full size “window,” so you can only see one application at any one time. However, you can move between them whenever you wish. The keys to do this (the *switch keys*) are defined on the next screen.

☐ Ctrl+Alt+Del global reset

If you select YES, pressing the **Ctrl+Alt+Del** keys together on the main console resets the complete system. The work of all users will be interrupted and information lost. If you select NO, then only the current DOS session gets stopped, and other DOS sessions on the main console and on other terminals are not affected.

☐ Status line enabled

Select YES if you want to have the status line displayed at the bottom of your screen. The status line shows the number of DOS sessions and the process running in each one.

☐ Mouse required

If you elect to use a mouse on the main console, another screen is displayed so that you can define the type of mouse.

☐ 84 key or enhanced (102 key) keyboard

Select the option that applies to your keyboard.

☐ National or US keyboard

If you have selected national language support in the previous screen, you now have the opportunity to confirm that choice for the main console keyboard, or revert to US English.

Enable/disable switch keys

The next two screens let you define the key combinations that will be used to switch between DOS sessions.

The operating system provides several ways of switching between DOS sessions. You can:

- ☐ Select a DOS session. This takes you straight to a DOS session.
- ☐ Cycle through DOS sessions. This moves you through the DOS sessions in sequence.
- ☐ Display a DOS session menu. This brings up a menu showing each DOS session and its associated process.

You can enable all three methods, or disable any of them. You must, however, have at least one method enabled.

Having enabled the switching methods of your choice, you are asked to confirm the default key combinations associated with them, or enter the combinations of your choice.

Note

If you enable both the second and third methods, be sure to select different key combinations for them.

System directory

You are now asked to define the directory into which the operating system will be installed. This is usually **c:\osutils**, but can be any directory on the hard disk. The directory you specify is created during installation if it does not exist.

(If you are running **SETUP**, you see a different screen, which asks you for the directory from which system files such as device drivers should be loaded. This is normally the directory into which you installed the operating system.)

Memory configuration

The next screens ask how you want to use the memory on your system. If you are unsure of the impact of any options, use the default settings. For more information about how memory is used, see Appendix D, "Memory management."

The first screen you see is for your information only. You are next presented with the following options:

☐ Maximum memory per process

This option allows you to specify the maximum amount of memory that can be allocated to any one process. Changing this value affects all users. If you want to limit the memory available to particular users or applications, use the PIFED or MEMSIZE commands at the command prompt or in a batch file.

☐ Additional system memory pool

This option lets you increase the pool of memory available to dynamic structures. There is usually enough space, but if a file such as a device driver will not load, allocate more memory using this option.

☐ LIM emulation enabled

Use LIM emulation only if you are running programs that require LIM expanded memory, for example large databases and spreadsheets. If LIM emulation is enabled, a 64 Kbyte page frame will be allocated.

☐ All upper memory set to autoscan

This option controls the way that the area of upper memory between the video memory and the top of the first megabyte is used.

If, in the third option, you enable LIM emulation, a subsequent screen asks you to specify further details:

☐ Page frame base address

This option lets you decide where and how the 64 Kbyte page frame should be created. Choose between Auto, TPA, and one of the displayed addresses.

The `Auto` and `address` options allocate 64 Kbytes of upper memory permanently for paging. In the former case, the operating system allocates the first four contiguous 16 Kbyte blocks it finds; in the latter case you specify the base address. Note that if any of the 16 Kbyte blocks within the 64 Kbytes page frame are unavailable, LIM emulation is automatically disabled. In this case, you will need to run `SETUP` to return to this screen and move the page frame base address to an area of memory that is free.

With either of these methods, the page frame uses up memory that would otherwise be used by the operating system itself, so that more of the operating system is left in conventional memory reducing the amount of memory available to applications. This affects all users.

The **TPA** option lets you move the 64 Kbyte page frame from upper memory into the part of conventional memory used for running programs (the TPA, or Transient Program Area). This lets you overcome the problems of using LIM emulation mentioned above. By placing the page frame in the TPA, the amount of the operating system that can be moved into upper memory is not reduced keeping system performance overall to a maximum.

The page frame is only allocated when a program requiring LIM is run, reducing the TPA by 64 Kbytes while the program is running. This will not affect other applications.

Note that you will need to run **PIFED** on all applications that use LIM if you select this option. (See Chapter 15, "Running applications," for more on **PIFED**.)

☐ **Maximum LIM memory per process**

This option sets the maximum amount of LIM memory that can be allocated to a DOS session. The default value is 1024 Kbytes. If you want to limit the amount of LIM memory for a particular application, use the **PIFED** or **LIMSIZE** command.

If, in the autoscan option, you select **YES**, all segments of upper memory are checked when next you boot up the operating system. If they are free, they are enabled and used for buffer space allocation. If they are not free, they are disabled.

If you select **NO** in the autoscan option, further screens allow you to set either the status of individual 16 Kbyte blocks of upper memory, or all upper memory to the same value. In both cases, the options are:

☐ **Disabled**

This means do not use this block.

☐ **Autoscan**

This means use this block the next time the operating system is started if it is free. The status will then be changed to enabled. If the block was not free, the status becomes disabled.

☐ Enabled

This means use this block even if it was not free. Blocks may be in use if you have network, screen driver or intelligent I/O cards installed. Normally this memory should be disabled (either directly or by autoscanning), but if the card is not being used, you may want to use the memory for buffer allocation.

WARNING If you are not sure which option to use, use Autoscan.

When all your memory configuration settings are to your liking, press **Enter**.

System options and parameters

The next series of screens ask about a variety of system options and parameters. The first screen you see is for your information only.

System options

You are presented with a screen containing the following options:

☐ Virtual disk support (VDISK.SYS)

This option allows you to use a part of the system's Random Access Memory (RAM) as a virtual disk.

Such a disk has a much faster access time than a physical disk, but the information stored in it is lost when the system is reset. If you want to save any information on the virtual disk, copy it to the hard disk or to a diskette before you reset the system.

If you select YES, a further screen lets you define the size of the virtual disk.

This option will create one virtual disk of the size you specify by creating a `DEVICE=vdisk.sys` entry in the **cconfig.sys** file. Note that the virtual disk is always allocated in expanded memory.

☐ Hard disk cache (CACHE.EXE)

Cache is an area of memory used to temporarily store information that is frequently read from disk. Applications then use the information in the cache, which is much faster than accessing it from a hard disk or diskette.

If you select YES, a second screen asks you to set the size of the cache.

☐ File Fastopen support

Fastopen support allows you to balance disk performance against memory requirements. It is used to access files more quickly than otherwise. This is done by keeping an index of the Fastopen files in memory (upper memory, if available). You specify the number of files on a second screen; in a typical system the default number of files (4096) would use 8192 bytes of memory.

Do not specify more files than necessary as this reduces the memory available to run programs and may prevent them loading.

☐ CSPOOL (Print Spooler)

The CSPOOL Print Spooler allows all your terminals to share the same printer without waiting until the printer is free. If you use spooling, then when you make a print request, a print job is created and queued. You can then carry on using the application even if the printer is not free to print that job. If spooling is not used, your DOS session will be suspended until printing has finished.

If you select YES, CSPOOL is installed, and each time you boot the system, spooling is started. For more information about spooling, see Chapter 13, "Printing."

System parameters

You are next presented with a screen containing the following options:

☐ Number of multigraphics memory/disk buffers

If you have a graphics terminal, the operating system allows you to run more than one graphics application simultaneously. When a graphics application is switched into the background, that is, when you move to another DOS session, the graphics image must be saved. This can be to

RAM or to a temporary file on disk. These two options define the number of buffers available to be allocated.

If a free memory buffer is available, the image will be saved in that buffer; otherwise a buffer is used on disk. Using memory is faster but uses up to 256 Kbytes of memory. Having multiple memory buffers therefore affects the amount of memory available for applications.

☐ Maximum open files per DOS session

This option specifies the number of files that can be open in each DOS session and is used to determine how much memory to set aside for controlling open files.

☐ Maximum total open files

This option specifies the total number of files that can be open across the entire system.

☐ Last drive

The operating system automatically assigns drive letters to the drives it finds during startup (including any virtual disks). It then sets the last drive as being the drive letter assigned to the last drive it finds. All the remaining letters become invalid.

This option allows you to set a different last drive, which you may need if you intend to run applications that require you to use a subdirectory as though it were a drive (using the SUBST command). Make the last drive the letter you will use.

Search paths

You are next presented with a screen asking where the operating system should look for:

- ☐ Commands, programs and batch files (the PATH).
- ☐ All other types of files (the APPEND statement).

You can enter more than one directory in each field. If you do so, you should separate them with semi-colons (;), for example, **c:\osutils;c:\osutils\cmdfiles**.

If you do not wish a PATH or APPEND statement in your **autoexec.bat** file, set the appropriate field to blank.

Additional options

You are next presented with a screen asking you to set certain additional options on or off. For more information on these options, see Chapter 16, “Customizing your system.” You are invited to set options for the following **cconfig.sys** commands:

☐ **NETDRIVE**

This enables file and record locking, preventing more than one user from updating a file or record at any time. This command is provided for specific network-aware applications.

☐ **IDLE**

This determines if idle detection is to be used globally.

☐ **SUSPEND**

This determines if a program is suspended when you switch to another DOS session.

☐ **BREAK**

This determines how **Ctrl+Break** stops a program. If you select **OFF**, the program is stopped only when it reads from the keyboard, writes to the screen or printer. With **ON**, the program is also stopped when it reads from or writes to disk.

Serial port configuration

The next set of screens let you set up the serial ports on the main console. The first screen is for your information only.

Change/add multiport card

You are now prompted to add or change a multiport card, then, unless you have selected **None**, given a menu of multiport cards to choose from. A further screen lets you define the configuration of the card.

Multiport options are as follows:

☐ Base port address

The base port address used by the card. This should correspond to the switch settings on the card itself.

☐ Number of ports

The number of ports supported by the card.

☐ IRQ number

This setting identifies the interrupt request line used. It should correspond to the switch settings on the card itself.

Port configuration method

You are now asked to establish the method used to configure the ports on the main console and any multiport card. Three methods are available:

☐ Automatic

Ports are allocated automatically once you have defined the number of ports for each function needed on your system.

Ports are grouped by function: PC terminals, mice for PC terminals, dumb terminals and printer/auxiliary ports. Each group is then configured and all the ports in that group have the same configuration.

This option is recommended if you are configuring a large system from scratch.

Caution The “automatic” route can save time if you have a large system. However, any configurations for individual ports already set up are lost and must be reentered.

☐ Standard

This option is similar to the automatic method in that ports of a given function are given the same configuration, but this option does not allocate port numbers automatically - all existing allocations are retained.

Use this option if your system is already configured and you need to change the configuration of a particular port function, for example, for all PC terminals.

☐ Configure individual ports

Each port must be configured separately.

This option is recommended if your system is already configured and you want to change the function of one or more ports. Also, use it if one or more ports of a given function require a different configuration from other ports of the same function.

Port configuration

You are shown a default allocation of devices to serial ports. Establish the correct numbers of each type of device, PC terminal, mouse, dumb terminal, and printer/aux port.

For each type of device that is represented in your system, you have the opportunity to set the following parameters:

- ☐ Baud rate
- ☐ Parity
- ☐ Data bits
- ☐ Stop bits
- ☐ Protocol

If you are not sure what values to set, consult the documentation that came with your terminals, mice or printers.

If you have identified PC terminals as being present in your system, you are prompted for further input, with a subset of the options available for the main console. (See the section "Main console configuration" earlier in this chapter for a description of these options.)

When you have completed your port configuration (or earlier if you have selected the `Configure individual ports` method), you are shown a summary screen listing the currently configured ports. For example:

Port	Function
1 COM1	Printer/Aux Port
2	Main console mouse
3	PC terminal (Station 1) with mouse on port 6
4	PC terminal (Station 2)
5	PC terminal (Station 3)
6	Mouse for PC Terminal on port 3 (Station 1)
7	Dumb terminal
8	Printer/Aux Port
9	Printer/Aux Port

Note that:

- ❑ Port 1 is COM1 on the main console. If you will be using a mouse for your main console it is possible to use this port.
- ❑ If you will be using a mouse for your main console and chose to use a port on the multiport card, port 2 will be used.
- ❑ PC terminals (such as a Wyse 60 or a PC running the terminal emulation program PCTERM) are allocated the next port numbers.
- ❑ The next numbers are used for any mice to be used with PC terminals. Note that mice should be connected to the multiport card ports - not to the PC terminals they will control.
- ❑ Dumb terminals (for example, terminals that do not support PC terminal emulation, such as a Wyse 50) and printers are allocated the remaining port numbers.
- ❑ As well as a port number, terminals are identified by a Station number. This number is displayed on the status line (or screen switch menu box) when you run the operating system. Bear in mind the station number of the terminal you work on, as it is used in some operating system commands.

Note

Make sure that any terminals are physically connected in such a way as reflects the configuration shown in the summary screen.

You can now select any port in the list for further configuration. When you are happy with all your settings, press **Enter**. This concludes the configuration part of the installation.

1.5 Installing system security

The operating system can be installed with or without system security. With system security installed, each user has a user name and password he or she has to type to gain entry to the system, and access to files and directories can be strictly

Create the Superuser

controlled. (For more information on system security, refer to Chapter 8, "Protecting your information.")

If you elect to install system security one special user, called *Superuser*, is created. Only the Superuser is able to run, edit or delete certain system utilities and files.

Select the **YES** option to install system security, the **NO** option to install an open system.

Renaming the Superuser and Supergroup

If you are installing system security, you can now rename Superuser and Supergroup. Press **Enter** when you are happy with the names.

Entering the Superuser password

If you are installing system security, type in the password that Superuser will enter to log in initially. Passwords can be from 6 to 12 characters long. (For full details on valid password characters, refer to Chapter 8, "Protecting your information.") Next, retype your Superuser password for confirmation.

1.6 Copying the system

The operating system is now copied from diskette to hard disk. When you have copied the first diskette in your distribution, you are prompted to insert the next diskette in the sequence. During the copy procedure:

- ☐ A special system file containing system values is copied to the root directory. If a file of this name already exists, it is overwritten.
- ☐ The file **command.com** is copied to the directory you specified (the default is **c:\osutils**).
- ☐ Two files, **autoexec.bat** and **cconfig.sys**, are created in your root directory. These files contain configuration information that the operating system needs to access every time it is started up. For more information, see Chapter 16, "Customizing your system." If **autoexec.bat** already exists on the disk, new commands are added to the file. If **cconfig.sys** already exists, it is overwritten.

- ❑ An ASCII input file, with the name of the system file and the **.inp** extension, is created in the root directory containing the system values. You can edit this file with any ASCII editor and use it to set up the system by running **SETUP** specifying this file as the input file. (For more information see Appendix E, “Using a setup input file.”)
- ❑ Security information files are written to a **security** subdirectory, if you have chosen to install system security.
- ❑ Certain CSPOOL files are written to a **spool** subdirectory, if you have chosen to install CSPOOL.
- ❑ All other files are copied to the directory you specified. If similarly named files already exist on these directories, they are overwritten.
- ❑ If a **read.me** file was supplied in your distribution, you are asked if you wish to read it. This file usually contains information relevant to a particular release.

1.7 Rebooting

When installation (or configuration) is complete, you are asked if you want to reboot, or Exit to operating system. Select the reboot option to start your new system up from the hard disk. Before doing so make sure there is no longer any diskette in your diskette drive.

After your computer has completed its startup tests, you see the operating system welcome screen. (If another DOS operating system is present, you are asked if you want to load this operating system. Enter Y and you see the welcome screen.)

You can always return any time in the future to alter any aspect of your configuration by using the **SETUP** program.

1.8 Logging in as Superuser

If you have chosen to activate system security, the first screen you encounter is a Login screen. You must log in to the operating system as the Superuser created during the

installation procedure. (Immediately after installation the Superuser is the only user on the system.)

Type the name and password established during installation. When the password has been checked, you see the system prompt, for example:

C>

You should not log in as Superuser for your everyday work. You need to create a user account for yourself, as described in the following chapter.

1.9 Copying your master diskettes

Before going any further, make a copy of your operating system master diskettes using the DISKCOPY command. The DISKCOPY command automatically formats a diskette for you if it is not already formatted.

WARNING DISKCOPY erases any information already stored on the diskette to which you are copying. Make sure you use a blank diskette or one whose contents are not important to you. You need to know your drive and diskette types before you copy diskettes. You cannot copy from one type of physical drive to another, for example from a 3 $\frac{1}{2}$ " to a 5 $\frac{1}{4}$ " drive, or between different density drives, for example from a 1.2 Mbyte disk to a 360 Kbyte disk.

To copy your master diskettes:

1 Type:

DISKCOPY a: a:

You are prompted to insert your source diskette. Insert the first installation diskette into diskette drive A:. This is your source diskette, the diskette you want to copy. Press any key to start copying.

2 A message appears tracking the copying process. You are prompted to insert your destination diskette. Remove the installation diskette and insert a blank diskette to which to copy. Press any key to continue.

- 3

 Follow the instructions as they advise you, inserting the source or destination diskettes in turn. Use the same source and destination diskettes until the DISKCOPY process is completed.
- 4

 When the copying is complete, you are asked:

Copy another diskette (Y/N)?

Remove the diskette already in the drive, press Y and insert the second installation diskette. Repeat the process until all the installation diskettes are copied.
- 5

 Label your copied diskettes clearly, write-protect them, and store them and your master diskettes in a secure place.

Twin-diskette-drive systems

If you have a twin-diskette system, insert the master diskette into drive A: and a blank diskette into drive B:, then type:

DISKCOPY a: b:

Follow the instructions on the screen.

1.10 Other tasks to perform

If you are using system security, the next thing you must do is establish user accounts for all the would-be users of the system. The next chapter describes how you do this.

Whether or not you are using system security, you may also need to address the following issues:

- ☐ Users on PCs acting as terminals need to run the PCTERM program on their PC before they can use the operating system. See Chapter 3, “Adding a PC as a terminal.”
 - ☐ You may need to reconfigure some applications so that they run in a multiuser environment. See Chapter 15, “Running applications.”
 - ☐ INSTALL and SETUP affect the complete operating system environment. You can tune the system for a particular DOS session, as described in Chapter 16, “Customizing your system.”
-
-

Setting up a secure system

THE operating system may be installed as an “open” system similar to a single-user DOS operating system, or as a “secure” system.

If you choose to have an open system, no user accounts or passwords are set up. Every user can use all the operating system commands, and the only restriction on file and directory access is provided by password protection.

On a genuine multiuser system, security is recommended, so that each user has to identify him- or herself to the system in order to gain access to it. System security allows you to restrict access to certain commands, and gives all users an opportunity to protect files that are private.

If you have opted for system security at installation time, you will have been given a special user account under the name of Superuser. This chapter introduces you, the Superuser, to your most basic responsibilities as system administrator. It tells you how to add other users and how to manage their accounts. It also shows how you can make a system more secure by restricting unauthorized access to hard disk partitions.

2.1 User accounts

When a secure system is booted up and the terminals switched on, each user sees a login screen. Each user must log in by entering his or her login user name and password. Only when the user name and password have been validated is the system prompt displayed.

From this point on, the user can access all DOS sessions on the terminal and work in the normal way. At the end of a working session he or she must exit from all programs and enter the LOGOUT command. This can be done from any DOS session. The login screen is redisplayed. (If the user needs to leave the terminal for some time but wants to leave programs running, the terminal can be “locked” using the LOCK command.)

Home directories

The system prompt may be the C> prompt, but is more likely to show the *home directory* defined by you as Superuser for that particular user. The home directory is the subdirectory in which each user begins a working session. It is in this directory, or in directories beneath it in the directory structure, that the user will keep files that require protection.

If an **autoexec.bat** file exists in a user’s home directory, it is executed at login time. Because it is executed after the **autoexec.bat** file in the system directory, any commands placed in this *local autoexec.bat* file override the global settings on the terminal that the user is working on. This can be useful where, for example, a user has a preference for a particular set of DOS session switching keys.

Each user is a member of a *group*. There can be up to 256 groups, each one containing up to 256 users.

Access permissions

Each file and directory in a secure system has an owner (usually, the creator of the file or directory). *Access permissions* define which users can use the file or directory and in what way. These access permissions are defined for the owner, the owner’s group and for everyone else (world access). Members of a group may be able to update or delete a file that other users cannot even read. The access permissions are defined in terms of read, write and delete access. For example, the owner will probably have complete access permissions (read, write and

delete) and may decide to let members of the group read the file. World access may be “none.” (For more information about access permissions, see Chapter 8, “Protecting your information.”)

Default permissions

The Superuser sets up default access permissions when a user account is created. Each user can then change these defaults.

2.2 Adding users

When you have installed the operating system with security enabled, log in using the Superuser name and password established during installation.

The Superuser account is only intended for system maintenance and should not be used for your everyday work. Your first task as Superuser should therefore be to create user accounts for yourself and your colleagues.

SECURITY command

To create a new user account, invoke the SECURITY command. This is a menu-driven utility, with options that let you add, edit, list and delete user accounts, and manage hard disk partition access.

Type SECURITY and press **Enter**. Follow the instructions on the screen. You can at any time press **F1** for full contextual help.

- ☐ Select the option to add a new user.
- ☐ Enter either the group name or the group identification number to which the user is to belong.

Each user must belong to one group, the name of which can be entered here. Alternatively, you can enter the group id (a hexadecimal number between 0 and FF). Group names can be up to 12 characters long and can contain any letters or numbers, along with underscore characters, hyphens and single quotes. Group and user names must be unique; for example, you cannot have a group **accounts** and a user **accounts**.

Note

User names, groups and passwords are case insensitive, so for example ACCOUNTS is taken to be the same as

accounts.

- ☐ Type in the login user name. This can be up to 12 characters long and can contain any letters or numbers, along with underscore characters, hyphens and single quotes. User names must be unique.

- ☐ The next available user id in the group is displayed.

Each user is given a unique identifying number (0 to FF). Accept the number displayed, or enter a different (but as yet unused) number.

- ☐ Enter the login password.

This is the password the user will enter to log in for the first time. The password must be from three to 12 characters long and can contain any letters or numbers, along with underscore characters, hyphens and single quotes. (Users should change the password with the PASSWORD command as soon as they have logged in for the first time.)

If you leave the password field blank, no password will be associated with the account, and users will be able to access it simply by entering the correct login name. This arrangement is only recommended for accounts that are intended to be shared.

- ☐ Enter the default access permissions that will apply to the files the user creates. Default access rights are usually `rwd rwd r--`. That is, the user can read, write to and delete the files in his or her ownership. Other users in the same group can do the same, but members of other groups are restricted to read-only permissions.

- ☐ Enter the user's home drive and directory.

This will be the user's current drive and directory whenever he or she logs in. The choice of home directory depends on how you want to organize the system. The default assigns each group a directory beneath the root directory of your hard disk, and each user a home directory within that group. For example, for user **derek** in group **accounts**, the home drive and directory **c:\accounts\derek** is offered as default. SECURITY creates the group and user directories you specify, if they do not already exist.

- ☐ Enter any brief comments you wish to make.

The details you have entered are displayed. If you wish to change any of the entries, press **ESC** to retrace your steps to the appropriate screen, then make your corrections.

When you are happy with your entries, select the option to add the information to the operating system database.

2.3 Maintaining user accounts

The SECURITY command allows you to list and edit details of existing user accounts. It also lets you delete user accounts (for example, when someone leaves your company).

For information on these options, invoke on-line help as follows:

```
C>SECURITY /H
```

2.4 Making the operating system partition secure

There may be a risk, despite the security furnished by user accounts and access permissions, of unauthorized access to the operating system partition, by way of another DOS operating system.

To render the files on your operating system partition unreadable from diskette or any other partition on the hard disk, select the appropriate SECURITY option. Normal use of the system should not be affected in any way by such a move. If you find otherwise, you can easily reverse the operation.

Note

You may have good reasons on occasions for wishing to access your operating system partition from another operating system. In such cases, use SECURITY to remove partition protection temporarily, then re-establish a secure partition when you no longer require this kind of access.

2.5 Making special files secure

A number of files are particularly deserving of the protection afforded by the system security features. These files are created by INSTALL with the recommended access permissions. In some cases, these allow read access to users other than the Superuser, who is designated the owner of all such sensitive files. In other cases, ordinary users are denied all access. Make sure that the established file ownership and access permissions are not modified at any time.

The following files, normally located in the root directory of the hard disk on which the operating system has been installed, should have the access permissions as shown:

rwd --- ---	cconfig.sys	(the system configuration file)
rwd r-- r--	autoexec.bat	(the system startup batch file)

The INSTALL utility also creates a directory called **security** beneath the system directory (normally **\osutils**). This directory contains a variety of files necessary for the support of user accounts. These files, and their associated access permissions, are as follows:

rwd --- ---	user.sys	(encrypted user passwords)
rwd --- ---	usersys.bak	(backup of user.sys)
rwd r-- r--	user.inf	(public user account details)
rwd r-- r--	userinf.bak	(backup of user.inf)
rwd r-- r--	login.log	(history of user login attempts)
rwd r-- r--	login.txt	(text displayed at login)
rwd r-- r--	login.err	(text displayed at failed login)
rwd r-- r--	lock.txt	(text displayed by LOCK)
rwd r-- r--	lock.err	(text displayed at failed unlock)

The **user.sys** file associates user id numbers with encrypted passwords. It is referenced whenever a user's password needs to be verified, most notably at login time. The file is updated when the user changes his or her password. If the file is deleted or becomes corrupt, not even the Superuser can log in. If for some reason the file does get lost, copy the **usersys.bak** backup.

The **user.inf** file is an ASCII text file containing user account information for each user on the system. This information may be read by all users, but should only be modified by the Superuser, preferably through the SECURITY utility.

Note that **user.inf** and **user.sys** must match. For example, if there is an entry for group and user id 1234 in **user.inf**, there must be a corresponding entry for 1234 in **user.sys**. If for some reason the **user.inf** file gets lost, copy the **userinf.bak** backup.

The **login.log** file records every act of logging in and out performed by all users. If you have no interest in this information, you can delete the file.

The **login.txt**, **login.err**, **lock.txt** and **lock.err** files contain message text, and any associated escape sequences and environment variables, used by the LOGIN and LOCK utilities. The two **.txt** files have fields for name and password, the **.err** files a field for an error message. By editing the text in these fields, you can create customized screen displays for LOGIN and LOCK. If any of these files are not present in the security directory, standard system messages are employed instead.

2.6 Making commands secure

You are recommended to make the following commands available to the Superuser only:

BACKUP	REBOOT
CHKDSK	RESTORE
DISKCOMP	SETUP
DISKCOPY	SID
FDISK	XSTOP
FORMAT	

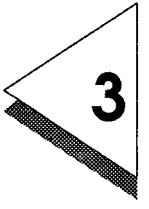
To do this, use the XATTRIB command to establish file access permissions such that other users cannot execute the commands. XATTRIB is described in Chapter 8, “Protecting your information.”

2.7 Words of advice

No operating system can be made one hundred per cent secure by software means. As system administrator, you should be aware that there is no substitute for physical security.

Where confidential information is involved, files should be stored on diskettes and those diskettes, when not in use, kept under lock and key.

Adding a PC as a terminal



3

THIS chapter describes the PCTERM terminal emulator program. When run with a DOS operating system on an industry-standard PC, it allows the PC to act as a terminal in a multiuser environment.

Note that you should have the correct hardware connections in place before you try to run the program.

3.1 Introduction

On a multiuser system only one user can use the main console at any one time. Often the main console will be reserved for the exclusive use of a single user, probably the person accredited with Superuser privileges.

Other users in a multiuser system can work on serial terminals attached by serial (RS-232C) communication cables to the main console. PCTERM enables PCs to be used instead of one or all of the serial terminals. These PCs can then revert to running local DOS programs when they are not required to act as terminals.

PCTERM requirements

PCTERM can be used on any industry-standard personal computer. The PC must have at least 256 Kbyte of RAM, a monochrome or color display and a serial port (either COM1 or COM2). The serial terminal cable must be plugged into the D-type serial connector on the PC. Depending on the type of PC, the serial port will be either a 9 or 25 way connector. Appendix B, "Connecting a serial terminal," gives details of the cabling requirements.

3.2 Running PCTERM

First make sure you have the hardware connections correctly set up. Then, working from the PC you wish to connect:

- 1** Copy the file **pcterm.com** from the second distribution diskette to a disk on the PC. If you have a hard disk on the PC, you are recommended to use that. (Alternatively, use the **FILELINK** command on an auxiliary port to transfer **pcterm.com** to the PC. For information about using the **FILELINK** command and cabling see Chapter 11, "Commands.")
- 2** Invoke the terminal emulator. To do so, type:

PCTERM

and press the **Enter** key.
The emulator now displays a menu with three options.
- 3** Press **F1** (**ON-LINE**) to start the emulator. The keyboard and screen are connected to the multiuser system and the PC is ready to start work as a terminal.
- 4** When you have finished using the multiuser system, press **Ctrl+Alt+F10** to redisplay the PCTERM menu.
- 5** Press **Esc** (**EXIT EMULATION**). You are returned to the PC's DOS command prompt and you can use the local DOS operating system or turn off the computer.

When you select the **F1** option to start working as a terminal, PCTERM checks whether the requested communication port is physically available. If the communication port is unavailable the emulation cannot proceed and an error message is displayed. Press any key to return to the menu and use the **F2** option (**SET TERMINAL PARAMETERS**) to check that the parameters are correctly set.

One of the parameters can be set so that PCTERM bypasses the menu and goes directly in to emulator (**ON-LINE**) mode when you start PCTERM.

Returning to the PCTERM menu

You can return to the main menu at any time by pressing the **Ctrl+Alt+F10** keys.

3.3 Setting terminal parameters

When you select **F2** from the PCTERM menu, the Configuration Options menu is displayed. This menu allows you to specify configuration details to suit your PC.

When PCTERM is used for the first time, a set of default values are displayed. When you save changes made in the Configuration Options menu, a file called **pcterm.con** is created to hold the new values. The next time that PCTERM is invoked it uses the saved settings. From then on, changes made update the **pcterm.con** file. (**Pcterm.con** is created in the same directory as **pcterm.com**.)

Options in the menu have several possible values. Press the function key corresponding to the option as many times as you need to display the value that you want to use. The options are explained in the following table:

Table 3-1 PCTERM Configuration Settings

Key	Description
Esc	RETURN TO MAIN MENU When you press the Esc key, you are prompted: Update the configuration file? (Y/N) Enter Y to create (or update) a file pcterm.con and save the displayed settings. Pressing any other key returns you to the PCTERM menu, without saving the changes you have made.
F1	COMMUNICATIONS PORT Two options are available, COM1 and COM2.
F2	TX/RX BAUD RATE This sets the transmission speeds for both sending and receiving to one of the following values: 75, 110, 300, 600, 1200, 1800, 2400, 4800, 9600 or 19,200 bits per second.
F3	PARITY Sets the parity bit to NONE, ODD or EVEN.

Table 3-1 PCTERM Configuration Settings (continued)

Key	Description
F4	WORD LENGTH The word length is always 8 bits.
F5	FLOW CONTROL MODE Data can be transmitted faster than it can be processed at the receiving end. Choose the method used to control the flow of data between terminal and computer to stop loss of data: DTR (Data Terminal Ready), XON/XOFF or NONE.
F6	COLOR SCROLL MODE This is only relevant when using the color adapter. There are three options, FAST, SLOW or VERY SLOW. If the FAST option is selected, “snow” will appear on the screen when it is being updated. The effect can be avoided by using the slower modes, but this adds a performance overhead and slows down the display.
F7	DISPLAY INITIAL MENU YES displays the PCTERM menu each time PCTERM is run. The NO option allows regular users who do not want to change the configuration details to go straight in to terminal emulation mode on loading PCTERM. If NO is set, the PCTERM menu can be displayed by typing Ctrl+Alt+F10 .
F8	WRAP-AROUND This option determines whether the cursor moves to the beginning of the next line (ON) or stays at the end of the current line (OFF), when a character is displayed with the cursor in the last column. OFF is the normal setting.
F9	Selects the default printer device.



Part 2

Using

What is Multiuser DOS?



4

DIGITAL Research Multiuser DOS is an advanced operating system designed to exploit the speed and power of the Intel 80386, 80386SX and 80486 microprocessors. It is a multiuser, multitasking operating system, so it can accommodate a number of users, each performing a number of tasks simultaneously. It is aimed primarily at small businesses and departmental work groups, where there is a requirement for shared information and hardware resources.

Multiuser DOS supports the full range of standard DOS applications, such as databases, spreadsheets and word processing packages. It also supports native applications developed specifically for such vertical markets as the medical profession, the retail trade, accounting and wholesale distribution. It has the look and feel of a standard DOS operating system, but is considerably more powerful and flexible.

This chapter explains in more detail the various features of Multiuser DOS that may be new to you.

4.1 Multiuser support

Multiuser DOS is a DOS operating system. Like all such operating systems, Multiuser DOS performs a range of standard housekeeping tasks for your personal computer, such as managing applications, storing and giving access to files of information, and controlling the operation of screen, keyboard and printer.

However, while standard DOS operating systems are designed to cater for the needs of only one user at a time, Multiuser DOS has the ability to support a number of different users, all on the same system. These users can all be working simultaneously, each with the power of the operating system at his or her disposal.

Adding terminals

To make your system multiuser, all you have to do is connect additional terminals to the serial ports of your personal computer, using RS-232C cables. If you use multiport boards, you can connect a significant number of terminals in this way. Of course, you always have at your disposal the central personal computer, or main console as it is called.

Terminals can be of the low-cost, serial variety (consisting simply of keyboard and screen), or they can be personal computers, made to act like terminals through the use of the terminal emulation package, PCTERM, provided for the purpose.

Resource sharing

Each Multiuser DOS user can at any time make full use of the hardware resources of the personal computer system, such as the hard disk or the printer. This makes for a considerable saving in hardware costs. Each user can also share applications and files of information. This not only saves expense but makes for a more ordered and cooperative way of working.

Where printers are concerned, it is clearly not possible for multiple users to use the same printer at exactly the same time. Multiuser DOS offers the next best thing, a print spooler

utility that manages print requests and sends jobs to print in an orderly fashion.

Where shared disks and files are concerned, Multiuser DOS offers a number of methods of maintaining the security of individual users' information. The most powerful method involves a system of login names and passwords, which users must enter in order to access the system.

Networking-in-a-box

Multiuser DOS systems offer the capabilities of local area networks (LANs), the alternative multiuser solution, without the requirement for costly hardware and software add-ons. Many popular DOS LAN applications run on Multiuser DOS just as they would on a LAN.

Expanding the system

More terminals or PCs can always be added to a system at a later date, as long as there are serial ports available. If more users need to be added than can easily be supported by a single system, DR Net, the Digital Research LAN software package, is available to merge multiple Multiuser DOS configurations into a seamless whole. Multiuser DOS also provides connectivity to the de facto industry-standard, the Novell® LAN system.

4.2 Multitasking

Each user in a Multiuser DOS configuration can, at any one time, be working on a number of different tasks. Using a technique known as time slicing, the operating system dedicates a brief amount of time to each outstanding task, on a round-robin basis. The speed of this operation is such that each task appears to be the only one running.

Multitasking means that no single application can monopolize any of the resources of the system. So, for example, you could be working on a spreadsheet, sorting a database, printing a file and receiving other files through a modem, all at the same

time. Meanwhile, other users on the system could be performing the same or different combinations of tasks.

Multiuser DOS deals with interrupts as and when they occur, thus guaranteeing real-time response where this is required (for example, with electronic point-of-sale applications).

DOS sessions

Applications usually tie up the keyboard and screen when they are running. So how can you run more than one application on a single terminal? The answer is by using different DOS sessions.

DOS sessions (sometimes known as virtual consoles) are like the different channels on a television set. Each time you switch DOS sessions, using a simple key combination, you give yourself in effect a new screen and keyboard.

At any one time, one DOS session is in the foreground. It alone can accept input from the keyboard and it alone can display on the screen. But in the background you have a number of other DOS sessions, each of which may be running its own application.

Providing an application does not call for keyboard input, it runs just as happily in background as in foreground mode. If it does require periodic input, you can always switch DOS sessions to bring it back into the foreground.

You have up to a possible eight DOS sessions on the main console and each of the additional terminals in your Multiuser DOS system. (The actual maximum for each terminal is left to the discretion of the person in charge of your system.)

4.3 Performance enhancement

Multiuser DOS offers multiuser, multitasking performance that is second to none. Using a method called Dynamic Idle Detection, it detects and eliminates wasted processor power.

DOS applications are generally processor-intensive, often demanding power when they are in fact idle, such as when waiting for input from the keyboard. Multiuser DOS is able to

identify this wasted power and make it available to the other applications on the system.

In a typical configuration with four users running DOS applications, Multiuser DOS can give over 80% of the processor's power to each application. This contrasts with the approximate best of 25% available with other operating systems.

4.4 Memory management

Multiuser DOS uses the standard 32-bit extended memory available with 80386- and 80486-based personal computers. There is no requirement for any other type of memory. Extended memory, residing as it does on the motherboard of your personal computer, is generally very fast.

Multiuser DOS uses dynamic memory allocation to assign memory to applications as required. This method is highly memory-efficient, so the same amount of memory allows a greater number of users to be supported than on other systems.

The LIM specification is supported, so DOS applications that employ this standard run without a hitch.

Memory map optimization allows each task the maximum memory possible. On terminals this can rise to as high as 640K. This is well in excess of the sort of memory available on a single-user DOS machine, but with Multiuser DOS it can be achieved by each application in a multitasking, multiuser environment.

4.5 Multiuser DOS utilities

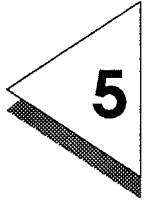
Multiuser DOS offers a superset of the utilities available with the usual DOS operating systems. For example, Multiuser DOS is there to help you:

- ☐ Enter and edit text
 - ☐ Manage and display files and directories of information
 - ☐ Copy files and disks
-

- ☐ Sort files
- ☐ Delete files
- ☐ Print

To aid novice users, help text can be called to the screen for each Multiuser DOS command. In many cases, lists of options, or menus as they are known, offer a further level of assistance.

Getting started



THIS chapter is your starting point for gaining hands-on experience of the operating system.

It starts with a brief explanation of some basic concepts. It then covers basic information on how to handle and write-protect diskettes, before moving on to describe the simplest operations you can perform on your personal computer.



5.1 Basic concepts

This section explains some concepts with which you need to be familiar before you begin using the operating system.

Disks

In order to hold all the information that you build up after working on a personal computer for some time, as well as the programs you use to help you in your work, you need some form of storage medium. Disks provide that medium.

Types of disk

When you start working with the operating system, you will be using two or three types of disk to store information:

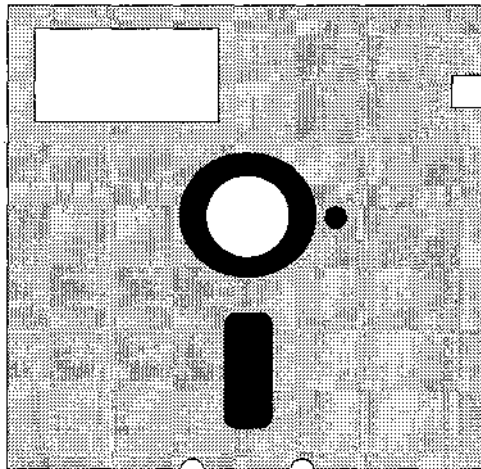
- ☐ Hard disks
- ☐ Diskettes
- ☐ Memory disks

Hard disks

Hard disks are permanently fixed inside your computer (they are sometimes referred to as fixed disks because of this). These disks can store a lot of information, and can access information much faster than diskettes. Hard disks come in many sizes, typically allowing between 20 and 300 Mbytes to be stored. Hard disks have to be partitioned before they can be used. Partitioning means dividing into sections. You then load your operating system into a partition. You can have several partitions on one hard disk.

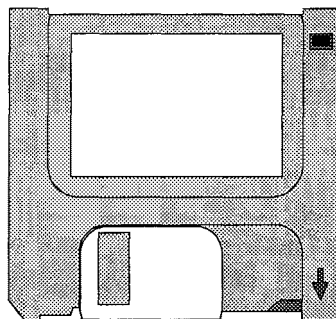
Diskettes

Diskettes (or "floppy disks") are removable disks that come in various sizes and capacities. The two main sizes are: 5¼" and 3½". The physical capacity of diskettes ranges from 360 Kbytes to 1.2 Mbytes for 5¼" diskettes, and for 3½" diskettes from 720 Kbytes to 1.44 Mbytes. New blank diskettes need to be formatted before you can use them.



5 1/4" diskette front view

The 3 1/2" diskettes are more compact than 5 1/4" diskettes. They can store as much as a 5 1/4" diskette or even more. They have solid plastic covers and a protective metal shield to prevent dust getting to the diskette surface.



3 1/2" diskette front view

Memory disks

There is a third type of disk, the memory disk (sometimes called a RAM disk or virtual disk). This is not a physical disk but part of the computer's memory that the operating system treats as a very fast disk. Files can be copied to or from a memory disk, and programs run from it. However, a memory disk is only a temporary means of storage, so when you switch off or reset your computer anything stored in the memory disk is lost. For permanent storage you must always use a hard disk or diskettes. Memory disks are not usually configured as standard. You can read more about setting up and using a memory disk in Chapter 16, "Customizing your system."

Drives

Disks are accessed by the slots or drives into which they fit on your personal computer. The various drives in your system are designated by letters followed by a colon. The first diskette drive is usually drive A:, and a second diskette drive drive B:. The main hard disk is usually referred to as drive C:. If you have multiple hard disk partitions, each partition is given its own drive letter, so it can be accessed like a separate drive. Memory disks are given the first available letter after the physical drives.

If you have a serial terminal to work at, instead of a personal computer, you will not have any disk drives actually attached to your machine. Instead, they will form part of the personal computer to which your terminal is connected. You can direct information to and from these disk drives, via their drive letters, just as if they were on your own desk.

Files and directories

Information that is to be held on a permanent basis on your computer is stored on disks, and is organized into files. A file can contain an application program, text like a letter or report, or computer data. A lot of your work with the operating system will involve files.

Files always have names to identify them (up to eight characters in length), and the filename can have an optional extension of up to three characters, separated from the filename by a period (.), for example:

letter.txt

The filename extension helps to identify the class of file; for example, **.txt** means the file is a text file.

Files are grouped together in directories. Directories keep information on the files they contain, including their size, time of creation, and when they were last changed. Directories can be arranged in a hierarchical fashion and so contain subdirectories as well as files. Files and directories are further explained in Chapter 7, "Managing your information."

Memory

Apart from the storage memory furnished by disks and diskettes, every computer must have memory to load and run applications and utilities. This type of memory is called Random Access Memory, or RAM. RAM can be thought of as the computer's workspace. The more complex an operation you want to perform, the more memory your computer needs.

Types of RAM memory

There are several types of RAM, including:

- ☐ Standard
- ☐ Extended
- ☐ Expanded

Standard memory is the memory that makes up the 640 Kbyte area available for applications on a single-user DOS system. With Multiuser DOS this amount of standard memory is available for each separate task.

Using the advanced memory management features of the 80386 and 80486 microprocessors, Multiuser DOS is able to allocate extended memory, which is memory above 1 Mbyte, to each task that requires it.

Another way of accessing more memory is to use expanded memory, otherwise known as EMS or LIM memory after the specifications that define it. Many DOS applications have been designed specifically to access this sort of memory. Multiuser DOS is able to convert extended memory into expanded memory as and when required, without demanding the addition of special expanded memory cards.

Appendix C, "Memory management," has more information on these and other memory types.

5.2 Using disks

Preparing disks for use

All physical disks (hard disks and diskettes) need to be prepared using the operating system before you can work with them. This process is called partitioning for hard disks and formatting for diskettes. In both cases it involves dividing the disk electronically into sections, so that the operating system can store and retrieve files more easily.

WARNING The preparation process erases everything stored on a disk, whether it be a hard disk or diskette, so great care must be taken when undertaking it.

Partitioning a hard disk

When you install the operating system, the INSTALL program detects whether your hard disk needs partitioning. You should only need to partition a hard disk once. In many cases your dealer will have already done this for you. Remember you should only partition your hard disk when it is new and blank or if you want to erase everything from it and start again. You partition a hard disk using the utility FDISK, as described in Chapter 11, "Commands."

Formatting diskettes

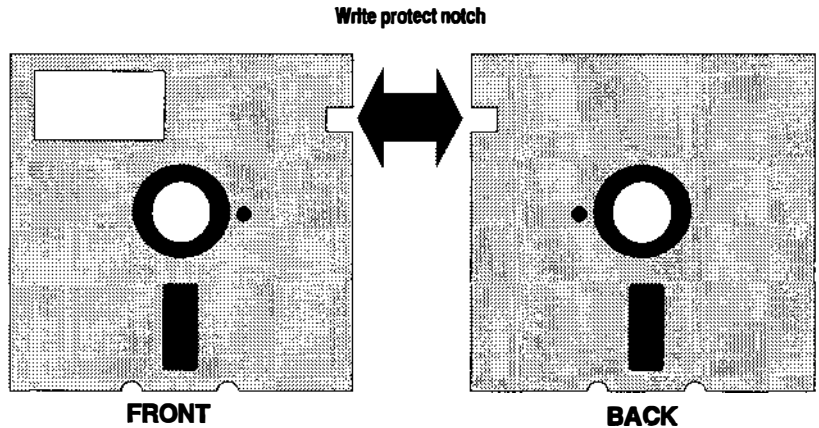
New, blank diskettes need to be formatted before you can store information on them. To format a diskette, use the command FORMAT. This command is explained in Chapter 11, "Commands." For FORMAT to work, your diskettes must not be write-protected.

Write-protecting diskettes

Diskettes can be physically write-protected to prevent anyone copying information onto them, or in any way changing their contents.

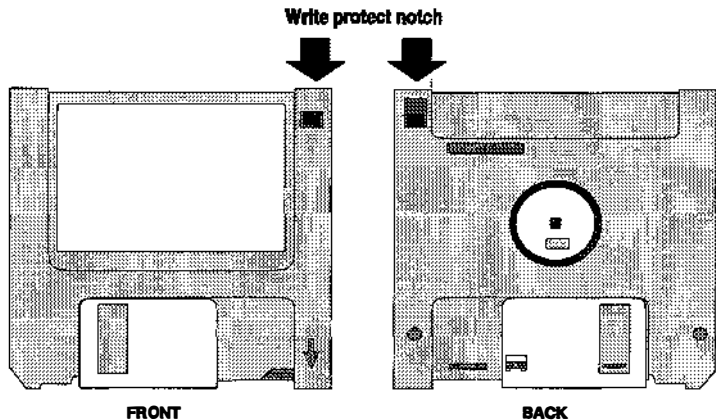
On 5¹/₄" diskettes, there is a notch in the top right-hand side of the diskette. To write-protect the diskette, you place a write-protect tab (usually supplied with your diskettes) over

this notch. If the tab is removed, the diskette is no longer protected. Write-protecting diskettes is a good way of ensuring that people realize a diskette has important contents.



Some 5 1/4" diskettes do not have the write-protect notch and these are permanently write-protected. Usually application diskettes come in this format to prevent you from erasing their contents.

With 3 1/2" diskettes, the write-protect notch is in the same place as a 5 1/4" diskette but the tab is built into the diskette. You slide the plastic tab up to protect the diskette; when you do this you can see a hole through the notch.



The do's and don'ts of handling diskettes

Do's:

- Do Hold diskettes at the corner.
- Do Use a felt tip pen to write the diskette label before sticking it onto the diskette, and if necessary remove the old label first.
- Do Store them in a suitable container in a safe place.
- Do Label diskettes clearly with what is recorded on them, including the name of the owner (if it is important) and the date it was recorded.
- Do Store diskettes in their protective envelopes.
- Do Take care when inserting diskettes into the diskette-drive. Slide them in without catching or bending them.
- Do Store at a temperature between 50 and 125 degrees Fahrenheit (10-52⁰ Celcius).

Don'ts:

- Don't Touch the inner diskette surface (the outer diskette cover can be touched).
 - Don't Keep diskettes near magnetic fields, radiators, heating or cooling ducts, or in direct sunlight.
 - Don't Place diskettes on top of your monitor.
 - Don't Let the diskette label adhesive near the exposed inner diskette.
 - Don't Bend or fold diskettes.
 - Don't Punch holes in the diskette using staplers or paper clips.
 - Don't Clean diskettes.
 - Don't Expose the inner diskette.
-
-

5.3 Logging in

In order to get started with your Multiuser DOS system, you must have the operating system installed correctly, and all the component parts of your hardware configuration switched on. The person in charge of your system, who may go under the title of System Administrator or Superuser, should have seen to it that your personal computer or terminal is fully operational.

On open systems

Now look at your screen. If it displays a letter followed by a > character, or a more lengthy series of characters, probably terminated by a >, you are seeing the system prompt. This implies you are part of what is called an open system, and have immediate access to all the facilities of the operating system. You can skip the rest of this section if you know you have an open system.

On secure systems

If on the other hand you see a message such as:

Please enter your login name:

this means that a system of user accounts is in place and you have more work to do to gain access to the operating system.

In this latter case the person in charge of your system should have already given you a login name and provisional password. Check if necessary what these are.

Now enter your login name at your keyboard. To do so, press the appropriate series of characters followed by the **Enter** (or **Return**) key.

You should now see a message such as:

Please enter your password:

Type in the provisional password you have been given, and press **Enter**. What you type will not show on the screen, but your input will be accepted none the less.

You should now see the system prompt on the screen, indicating that you have gained access to the operating system.

Note

Chapter 8, "Protecting your information," has more on login names and passwords, including information on how to change your provisional password.

5.4 Getting started at the command line

The system prompt

When you begin working with the operating system you start at what is known as the system prompt. Whenever you see the system prompt you know the computer is waiting for you to enter instructions at your keyboard.

If you have an open system, the prompt is likely to be:

```
C>
```

This means that drive C: is the current disk drive. When you enter an instruction, the operating system always looks by default to the current disk drive.

If you have a system of user accounts in place, and have had to log in, the prompt is likely to be something like:

```
C:\USER\FRED>
```

In this case the prompt shows not only the default drive, but the subdirectory where you are initially positioned. This is known as your home directory, and indicates the place allotted you on the system for your personal files. (For more on directories and subdirectories, see Chapter 7, “Managing your information.”)

The command line

The command line is the blank line to the right of the system prompt on your screen. It is here that any instructions you give the operating system will appear. The flashing cursor after the prompt is there to remind you exactly where the next character of input from your keyboard will show on the screen.

Changing between drives

If you want a command to be directed at a different drive, you change to it by typing the new drive letter plus a colon and pressing **Enter**. For example:

```
C>A: ↵
```

changes the current drive (C:) to diskette drive A:, as follows:

A>

Any instruction you type now is directed at drive A:. For example:

A>DIR ↵

gives a listing of all files and directories on a diskette inserted in drive A:.

You can tell the operating system to look at another drive without changing the current drive. To do this include the drive letter and colon when entering the command. So, with the example above you type:

C>DIR a: ↵

When giving instructions to your computer, for example starting a program, you must always direct them at the drive where the program or information is stored.

Typing instructions

After you enter an instruction at the command line, you must always press **Enter** to make the operating system carry out the instruction.

If you make a mistake when typing a command, you see a message like this:

```
Command or filename not recognized
```

It may be that you have merely mistyped the instruction or directed the command at the wrong drive.

If you make a typing error and you have not yet pressed **Enter**, you can correct it by pressing the **Backspace** key (←). This erases the command line one character at a time. You can then retype the command.

The operating system also has a powerful feature which stores previously entered commands and lets you recall them by pressing the up and down arrow keys. You can then edit and re-issue them. This and other ways of editing the command line are explained in Chapter 11, “Commands,” section 11.2.

Some simple commands

The operating system contains an extensive set of commands. A command is simply a method of getting your computer to carry out a particular task like copying a file. Most commands act upon whole disks, or a specified file or group of files.

Command names are up to eight characters in length and are usually followed by parameters and switches which qualify their operation.

One useful command which you can get to know now is DIR, which provides a DIRectory listing of files and directories. Try typing the letters DIR at the command prompt, and press **Enter**:

```
DIR .
```

You should see a listing of all the files and subdirectories in the current directory, with their size in bytes (characters) and the date and time they were last modified, plus how much space is left on the disk.

You can add switches to the DIR command, for example /W which shows the listing in a wide format. The sizes and modification times of the files are not displayed, but many more files can be listed on the screen.

Another useful command is the TREE command. It is similar to DIR in that it displays a listing of directories, but it shows all the directories on a disk, no matter where you are in the directory structure when you invoke it.

For more information on directories, see Chapter 7, "Managing your information."

Other useful commands include the SHOW command, which gives a status report on disks in the system, and LABEL and VOL, which let you assign a name to a disk and subsequently display it.

Getting help on commands

A special /H switch is available with all the operating system commands. This option displays help text on the command when you enter it, at the system prompt, after the name of the command itself. You can also use /? to achieve the same result.

For more information on commands, see the next chapter, “Working with commands.” Chapter 11, “Commands,” provides a reference section for all the operating system commands.

Running applications

To run an application, all you generally need to do is type its name on the command line, then press **Enter**. Refer to the application’s documentation for more information.

For more technical advice on running applications under this operating system, see Chapter 15, “Running applications.”

5.5 Switching DOS sessions

The prompt you get on your screen when you first gain access to the operating system, along with the command line that goes with it, are exclusive to a single DOS session. This is DOS session 1, always the first DOS session you have available to you at start-up time.

To switch to another DOS session, you can use any one of the following methods:

- ❑ Press **Ctrl+n**. That is, press the **Ctrl** key, along with a number on your keyboard’s numeric keypad. The numeric range is from one to eight (or less, depending on how your system has been set up).
- ❑ Press **Alt+Esc**. This moves you to the next DOS session available to you (initially, DOS session 2). Repeated use of this key combination will cycle you through the complete range of available DOS sessions.
- ❑ Press **Ctrl+Esc**. This displays a menu, from which you can select the DOS session of your choice.

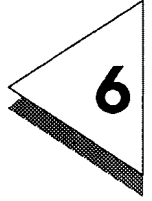
Note that these key combinations are only the defaults, and may already have been altered, or disabled, by the person in charge of your system. The **KEYS** command lets you see what your settings are.

These key combinations work even if you have an application program running in your foreground session. You do not need to have the system prompt on your screen.

If any of these key combinations conflicts with a key sequence used by an application, it can be changed by means of the SETUP utility (see Chapter 16, “Customizing your system”). SETUP lets users change switch keys on a system-wide basis. Alternatively, you can use KEYS to change switch keys for an individual terminal.

In each of your DOS sessions you can perform all the operations described in this chapter, and many more besides. This is the beauty of multitasking.

Working with commands



THIS chapter gives you some additional background material on the operating system commands.

It describes the various types of command there are, before highlighting some of the ways of invoking commands and controlling their output.

6.1 Command types

Internal commands

Certain commands that are particularly useful, and therefore needed often, are stored in memory for quick access. These commands are called internal or resident commands (because they reside in memory at all times). These commands are available to you whenever you have the system prompt in front of you on the screen.

You do not need to specify a particular drive or directory to use internal commands.

The internal commands are as follows:

Table 6-1 Internal Commands

APPEND	DIR	PATH	VER
ASSIGN	ERAQ	PROMPT	VERIFY
BANK	ERASE (ERA)	RENAME	VOL
BREAK	EXIT	RMDIR (RD)	
CHDIR (CD)	IDLE	SET	
CLS	LIMSIZE	STOP	
COPY	MEMSIZE	SUBST	
DATE	MKDIR (MD)	SUSPEND	
DEL	MORE	TIME	
DELQ	NETDRIVE	TYPE	

For a summary of what each command does, see the beginning of Chapter 11, "Commands."

External commands

Some commands that are used less often are not automatically loaded into memory. To save space they are stored on disk instead. These are external commands. They have the file extension **.com**, **.cmd** or **.exe**. When you enter an external

command, the operating system retrieves the “command file” from disk and runs it. The memory used by the command is freed when it has finished its job. BACKUP, RESTORE and DISKCOPY are examples of external commands, and will not work unless the operating system has access to a directory containing the command files.

The operating system provides a command called PATH to make sure that external commands can always be found, even if they are stored somewhere other than the current directory. The PATH command is described in the next chapter, “Managing your information.”

The external commands are as follows:

Table 6-2 External Commands

ATTRIB	FDISK	PRINT	TOUCH
AUXPORT	FILELINK	PRINTER	TREE
BACKUP	FIND	PRINTMAP	XATTRIB
CACHE	FORMAT	REBOOT	XCOPY
CHKDSK	KEYB	RECOVER	XDEL
COMMAND	KEYS	REPLACE	XDIR
COMP	LABEL	RESTORE	XSTOP
CSPOOL	LOCK	SECURITY	
DISKCOMP	LOGOUT	SETUP	
DISKCOPY	MAIL	SHARE	
EDITOR	MODE	SHOW	
EXE2BIN	PASSWORD	SID	
FASTOPEN	PCTERM	SORT	
FC	PIFED	SYS	

Batch processing subcommands

Batch files are files in which you store a set of often-used commands. When you type the name of the batch file at the command line and press **Enter**, the commands are carried out as if you had typed them in one after the other. For example, you can create a batch file which stores the commands you use to start an application like a spreadsheet.

Batch files can be very powerful and the operating system has a special set of subcommands for use in them. These subcommands are stored in the same way as internal commands.

Batch processing subcommands are listed below in Table 6-3. They are also described in more detail in Chapter 10, “Batch processing.”

Table 6-3 Batch Processing Subcommands

@	ECHO	IF	PAUSE
:LABEL	FOR	INPUT	REM
CALL	GOTO	INPUTC	SHIFT

Cconfig.sys commands

Apart from the commands already mentioned, there is a special set of system configuration commands which are not entered at the keyboard, but are put into a file called **cconfig.sys**. The **cconfig.sys** file is looked at by the operating system every time it is started. It then carries out any of the commands it finds there. For more information see Chapter 16, “Customizing your system.”

6.2 Entering commands

Commands consist of command names up to eight characters in length. These are usually followed on the command line by parameters that define the way the command is to operate, for example:

```
SHOW /D
```

Command switches

Command parameters shown with a forward slash, for example /D, are referred to as switches. This is because they are used to tell the operating system to switch to a particular variation on a command. /D tells SHOW to display the capacity and free space available for each disk drive in your system. /M on the other hand tells SHOW to display your memory configuration.

Typing the name of a command followed by /H displays help text on the screen. /? achieves the same results.

Punctuation is usually needed to separate command names and their options. Generally you put a space between a command name and an option, and between options themselves.

Sometimes other symbols must be used, for example commas or semi-colons; this is true of the MODE command:

```
MODE COM1:9600,N,8,1
```

Command syntax

The syntax or format of a command is made up of the command name itself and all its possible parameters and associated punctuation marks. In Chapter 11, “Commands,” each command description begins by setting out the full syntax. For example, the syntax for TREE, a useful command for getting an idea of the hierarchical tree structure of a disk, is shown as:

```
TREE [/H] [d:][wildspec] [/B] [/F] [/G] [/P] [/U:name]
```

The square brackets mean that the parameters they contain are optional. You are not expected to enter all the possible parameters at any one time; indeed, in the case of TREE to do so would make no sense.

You can enter commands and parameters in upper or lower case or a combination of both. Commands only take effect after you press the **Enter** key.

Specifying drives

Often, when entering a command, you need to state a source drive and a destination drive. The term drive refers to your hard disk and diskette drives, or it could refer to temporary memory disks. Remember that commands need to be directed at the correct disk drive for the operating system to be able to carry out the instruction. For example, with the BACKUP command (see Chapter 8, “Protecting your information”) you specify where you are backing up from and to:

```
BACKUP source_drive [filename] destination_drive
```

The source drive is generally your hard disk (for example C:), and the destination drive a diskette in drive A:.

6.3 Displaying data a screenful at a time

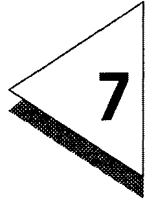
When a command displays a lot of information, it may scroll down the screen in such a way that you miss the first part of the display. This will happen for example if you enter a DIR command and the directory listing is so long it cannot all be shown on the screen at once.

You can stop a display from scrolling by pressing **Ctrl+S**. To start scrolling again press **Ctrl+Q**. Some commands, for example TYPE, have a /P switch which, if specified on the command line, causes listings to display one page at a time.

6.4 Stopping commands

You can stop commands when they are running by pressing **Ctrl+C** or **Ctrl+Break**. The processing of the command is stopped and the command prompt returned.

Managing your information



AS you work with your applications, you create your own information which needs to be organized for ease of access and security. You will also at some stage wish to perform certain operations on this information, such as copying, renaming, printing and deleting.

This chapter explains the tools that the operating system provides to help you manage your information. It concentrates on:

- ☐ Files
- ☐ Directories
- ☐ Paths
- ☐ Various commands that handle directories and files.

7.1 Storing data in files

Information on your computer is stored in files, and a lot of your work will involve them. A file is a collection of information that has something in common, for example it can consist of an application program, text like a letter or memo, or computer data.

Naming files

Files always have a name to identify them (up to eight characters long) and filenames often have an extension (up to three characters long, preceded by a period), for example:

myfile.doc

Your filenames can be any combination of the characters A-Z and 0-9. The following special characters are reserved by the operating system and must not be used in filenames:

Reserved characters

< > = , ; : . * ? [] () / \ + ! ~

Try to develop a logical system for naming your files, and make use of filename extensions. The filename extension should help you identify what kind of a file it is. For example, you can identify all your correspondence by giving each file an extension like **.let** for “letter” or **.doc** for “document.” Naming related files in this way becomes particularly useful when you want to handle groups of files using wildcards, which are discussed later in this chapter.

Some common filename extensions are explained in Table 7-1. You should not use these filename extensions for any other purposes.

Table 7-1 Common Filename Extensions

Extension	Description
.\$\$\$	Temporary or incorrectly-closed file. Many programs write temporary files to disk.
.tmp	
.bak	Backup file created by a text editor.
.bas	File containing a BASIC program.
.bat	File containing one or more commands to be executed sequentially (a batch file).
.cmd	Program file.
.com	DOS program file.
.exe	DOS program file.
.sys	Device drivers, the cconfig file and system files.

Filenames you should not use

The system uses special names to identify peripherals (devices added to your computer, such as printers). These reserved device names, listed below, should not be used as filenames.

Table 7-2 Reserved Device Names

Device name	Explanation
CON	User console (the keyboard and the screen).
AUX or COM n	Serial communications port.
PRN or LPT1	First parallel printer LPT1.
LPT2	Second parallel printer.
LPT3	Third parallel printer.
LST	Printer device.
NUL	All output to this device is discarded. No input is returned from the device.
CLOCK\$	Provides data and time support.

Handling groups of files (using wildcards)

To help you handle groups of files with ease and flexibility, there are two special characters, the question mark (?) and asterisk (*), known as wildcards.

Wildcards provide a means of referring to more than one file at a time in a command specification. Using wildcards you partially specify a file so that several files can match it. You can have wildcards in both the filename and the extension.

The ? wildcard

The question mark wildcard matches with any character in the exact position the question mark occupies.

For example:

`chapter?.doc`

matches with:

chapter1.doc
chapter2.doc
chapter3.doc

So, to copy all three chapters to a diskette, you only need the one command:

`COPY chapter?.doc a:`

The * wildcard

The asterisk wildcard is shorthand for several question mark wildcards. Where the question mark wildcard is only valid for the one position, the asterisk wildcard is valid from the position in which it appears to the end of the filename or extension.

For example:

`chap*. *`

matches with the chapter files:

chapter1.doc
chapter2.doc
chapter3.doc

However it also matches with other filenames like:

chapmem.fax
chaplist.doc
chapfax.bat

Commands and wildcards

The commands most often used with wildcards in their specification are COPY, DEL, ERASE, DIR, RENAME and TREE.

WARNING Wildcards are very powerful tools. They are extremely useful, but they can also be very destructive. For example *.* means all files. This is called the global wildcard. So if you type DEL *.* it erases ALL files in the directory in which it was carried out. When deleting files using *.* , you are always prompted before the operating system carries out the operation, but you should still take great care when using the global wildcard. If you want to be prompted before deleting any file, use the ERAQ or DELQ commands.

Handling groups of files (using filelists)

If a filename is preceded by the @ character, the operating system looks in the file for a list of filenames (a *filelist*). Some commands, that usually operate on only one file per command line, can thus be made to operate on several.

You create a filelist as a text file (using EDITOR or a word processor). You can enter as many filenames as you wish; you can even use wildcard characters in the names.

Note that each filename in your text file must be placed on a new line. Note also that you cannot use wildcard characters in the filelist name.

For example, you could create a file called **source.fl**, containing the names of the following files:

```
*.doc  
sample.txt  
henry.bak
```

If you then wanted to give one password to this group of related files, instead of issuing the command afresh for each file, you could type:

```
PASSWORD @source.fl /R:mycode
```

This assigns the password **mycode** to the files ***.doc**, **sample.txt** and **henry.bak**, in the subdirectory from which you gave the command. The @ symbol tells the PASSWORD command that the filename specified contains a filelist.

7.2 Arranging files in directories

The more files you create, the harder it is to keep track of them. You will find you need to organize them into groups according to their content, by project names, user names or classes. You can do this by using directories to order your files.

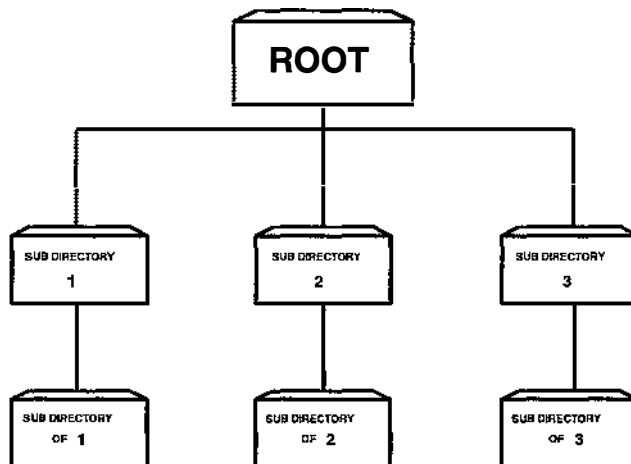
The root directory

When you prepare a disk or diskette and start storing files on it, an all-encompassing directory is automatically created. This is known as the root directory. Often, with diskettes, the only directory you will need is this root directory.

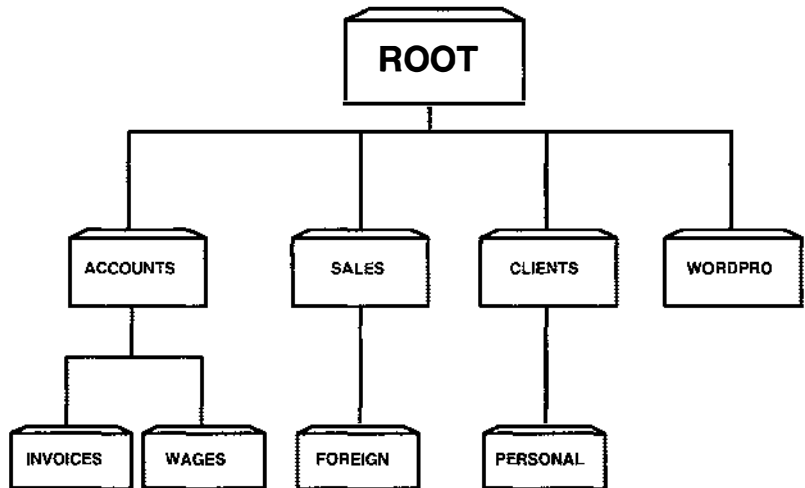
On a hard disk, the root directory is poorly equipped to contain all the files stored on it. Root directories can only contain a limited number of files and, besides, as the numbers of files grow, locating any one of them in a flat directory structure becomes well-nigh impossible. For these reasons, the operating system allows you to build up hierarchical directory structures, one for each disk (or disk partition) in the system.

Subdirectories

A hierarchical structure involves creating subdirectories beneath the root directory. Subdirectories can themselves contain further subdirectories. As you add more directories to the structure it will begin to grow into an inverted tree, with the root at the top and the branches growing downwards. For example:



In the following tree, four subdirectories were made from the root directory, **accounts**, **sales**, **clients**, and **wordpro**:



In the subdirectory **accounts**, two more subdirectories were created called **invoices** and **wages**. The **sales** and **clients** subdirectories also have subdirectories, **foreign** and **personal**.

Files can be created anywhere in the structure. You can create many levels, but it is not practical to make too many, as it then becomes difficult to locate your files.

Naming directories

Naming directories follows the same conventions as naming files: a one-to-eight character name plus an optional extension of up to three characters. Normally it makes sense to keep directory names short.

The current directory

The directory that you are working in at any one time is known as the current directory. When you carry out a task like creating a file, the operating system assumes you mean it to be created in the current directory (that is the directory in which you are currently working), unless you specify otherwise.

You can travel around the directory structure from the root downwards, or from anywhere within the structure back up to the root. You can also go straight to a directory without going through intervening directories. You move between directories using the CHDIR (change directory) command, described later in this chapter. Each time you change directories using CHDIR, the destination directory becomes the new current directory.

Usually, your system prompt tells you what your current directory is. However, if your system prompt is not set up to do this, you can always type CHDIR, without any parameter, to find out.

Special directory symbols

All directories (except the root) are contained within other directories. A directory which contains a subdirectory is known as the parent of the subdirectory. Special symbols indicate the current directory and the parent of (or directory above) the current directory:

- . Means the name of the current directory.
- .. Means the directory above the current one.

These entries are automatically created when you make a directory. When you issue a DIR command, the entries . and .. appear at the top of the listing.

If you type the command:

```
DIR ..
```

you get a directory listing of the files in the parent directory of your current directory.

You cannot delete the . and .. entries.

Your home directory

If your system has user accounts established, you probably find when you log in that you are not in the root directory of the default drive, but a subdirectory.

The directory in which you are initially placed when you log in is known as your home directory. Typically it will have a name that associates it with you and you alone.

Home drives

You are not obliged to remain in your home directory, though you are expected to keep information personal to you there, or in further subdirectories descending from your home directory.

You may also be able to refer to your home directory by a form of shorthand, that allows each user to identify his or her home directory by simply typing a drive specification. Typically, H: is used as this "home drive;" it is unlikely to refer to a real hard disk. The advantage of having a home drive set up in this way becomes apparent when you run applications.

The person in charge of your system can tell you if a home drive is in place for your account. If you wish to set one up for yourself, you must put a SUBST statement in your **autoexec.bat** file. (See Chapter 16, "Customizing your system," for more on the **autoexec.bat** file.)

7.3 Specifying paths

Within a tree directory structure, it is quite acceptable for more than one file or directory to have the same name. For example, there may be a file called **france.doc** in the directory **\sales** and in the directory **\accounts**. You need a way of referring to files and directories which allows them to be specified uniquely, so you are able to access each of the two **france.doc** files in the above example without confusion.

The operating system provides such a mechanism. Every time you want to access a file or directory you specify a path to it. A path is like a roadmap, it tells the operating system the route to follow to get to a particular file or directory.

You specify a path with a series of directory names, separated by backslashes (\). This traces a route through the tree structure to the desired point.

If the first directory in the path is preceded by a backslash, then the path begins at the root directory. Paths that begin with a directory name are taken as starting from the current directory.

For example:

```
c:\sales
```

In this path, the search is to start on the C: drive.

In the following path, **sales** and **foreign** indicate the names of the directories the operating system has to pass through to locate the file **france.doc**:

```
c:\sales\foreign\france.doc
```

File specifications

When you add a filename to a path it becomes a file specification (or filespec), in other words a way of specifying where a particular file is. See the first section of Chapter 11, "Commands," for more information on defining a file specification.

The path in a file specification can contain any number of directories, and can be anything up to 63 characters in length.

The PATH command

In the previous chapter mention was made of external commands, the commands which are stored on disk rather than in memory. They include all files with extensions **.cmd**, **.com** and **.exe**. The operating system needs to be told where these external commands are. You tell it how to locate them by using the PATH command.

The PATH command can be used to specify a series of paths that the operating system should search to find command program files. For example:

```
PATH=c:\osutils;c:\program;
```

With this command you are telling the operating system to first look in the subdirectory **c:\osutils** and then, if this proves unsuccessful, to look in the subdirectory **c:\program**. Note that the individual paths are separated by semi-colons (;).

Usually you would keep your most often used directories at the beginning of the path for speed of access.

Looking for external commands

It is good practice to keep your external commands in one directory. This is done automatically for you during installation; the external commands are placed in the directory **\osutils**. Therefore it is useful for you to have the directory **\osutils** at the beginning of the PATH:

```
PATH=c:\osutils
```

Once a path is set up, you can type commands without having to worry about being in the wrong directory. The path remains

set up until you restart your computer or issue another PATH command.

You can always display the current path by typing:

```
C>PATH
```

You can clear the path by typing:

```
C>PATH ;
```

As well as looking for external commands, you also use PATH to search for batch files (see Chapter 10, “Grouping commands in batch files”).

Using paths you can make the operating system search on different disks or diskettes for files or programs. You can set up a path like this:

```
PATH=c:\;a:\;
```

The system knows to look first in the root directory of the C: drive, and if the program is not there, to switch to the A: drive.

Note

Always specify the complete path including the drive for all the subdirectories in the path.

Looking on different disks

PATH and the autoexec.bat file

The best time to invoke the PATH command is as soon as you gain access to your system. You can do this by including a PATH command line in a file called **autoexec.bat**, which is always loaded as part of system startup. You can use the SETUP program to establish (or modify) your path settings, as described in Chapter 16, “Customizing your system.”

There can be different **autoexec.bat** settings for each virtual console, as well as each user, in the system. So the PATH settings can be different from one activity to the next.

Note

Another command that works in a similar way to the PATH command is APPEND. Instead of external commands, it works with data files. See the explanation for APPEND in Chapter 11, “Commands.”

7.4 Working with directories

Listing directories (TREE)

The TREE command allows you to list all the directories on a given drive and keep a track of your tree structure, for example:

```
C>TREE a:
```

gives you a listing of the directories on the A: drive.

The list gives the space used and the number of files belonging to each directory, as well as the path to each directory.

The TREE command has some useful parameters such as /F, which lists all the files in each directory, as well as the directories themselves. You can also produce a graphical display by specifying the switch /G.

Note

Another command which lists the contents of directories and their subdirectories is the XDIR command, or Extended Directory command, as described in Chapter 11, "Commands."

Making a directory (MKDIR)

To create a directory, use the MKDIR command (which can be abbreviated to MD). The directory above the one you wish to create must already exist.

For example, to create the directory **accounts**, type:

```
C>MKDIR \accounts
```

The root directory already exists, so do not try to create it. To make a directory below **accounts**, you can next type:

```
C>MKDIR \accounts\invoices
```

or you can create a directory from the **accounts** directory itself.

The directories, **accounts** and **invoices**, are now ready to receive files, and to be made the current directory. You can now create files with a word processor to store in these directories.

Note

Remember that the `.` and `..` entries in a directory are automatically created. The operating system uses these symbols to indicate the current directory, and the parent of the current directory.

Displaying and changing directories (CHDIR)

To change between directories, type **CHDIR** (or **CD** for short) followed by a path. The directory you change to becomes your current directory.

For example, to make the **accounts** directory the current directory type:

```
C>CHDIR accounts
```

To return to the root directory type:

```
C>CHDIR \
```

(remember that the backslash represents the root directory).

Also useful is the `..` symbol. When you type:

```
C>CHDIR ..
```

you are moved up one level, for example from the **accounts** directory to the root directory.

Note

If you type **CHDIR** without a path, you can find out the name of the directory you are in.

Removing a directory (RMDIR)

The **RMDIR** (or **RD**) command removes an empty directory. The format of the command is identical to that of **MKDIR** and **CHDIR**.

For example, to remove the subdirectory **\accounts\invoices**, where the current directory is **\accounts**, type:

```
C>RMDIR invoices
```

To remove the same directory when the current directory is the root, type:

```
C>RMDIR \accounts\invoices
```

Note the following about the RMDIR command:

- ☐ The directory to be deleted must be empty of files. The DIR command displays only the entries . and .. when a directory is empty. If DIR lists any more entries, the directory is not empty.
- ☐ The directory to be deleted must not have any subdirectories below it. Remove them first.
- ☐ You cannot delete a directory while it is the current directory.

7.5 General actions on files

This section touches on other useful commands you can use with files. Note that these are overviews; you should go on to read Chapter 11, “Commands,” for more detailed descriptions.

Copying files (COPY)

A command that you will need to use often is COPY. Using file to file or between directories.

For example:

```
C>COPY wages.txt a:
```

When you copy from disk to disk, you can use the same name for a file.

However you cannot do this on the same disk, unless the file is in a different directory. For example, it would be possible to type:

```
C>COPY report.doc c:\sales\report.doc
```

Note

The XCOPY command is another useful command for

copying. It can copy groups of files including a whole tree structure of files (for example files and directories) and is faster than COPY.

Deleting files

Eventually you will want to delete files from your disks or diskettes to keep them from becoming too cluttered. You can use one of the five delete commands to do this: DEL, ERASE, DELQ, ERAQ, or XDEL. Type the command followed by the filename you want to erase:

```
C>DEL myfile.txt
```

All the commands work in much the same way, except that DELQ and ERAQ always prompt you before deleting a file. XDEL enables you to delete files and subdirectories.

Caution Take care when using the delete commands, particularly if you use wildcard characters. If you want to use wildcard characters, it is safer to use the ERAQ or DELQ commands.

Displaying file contents (TYPE)

You can display the contents of a text file on your screen using the TYPE command like this:

```
C>TYPE myfile.txt
```

This command works only with text files. If you try to display a program file using TYPE, you get a lot of unintelligible characters and symbols.

If you are looking at a large file, and wish to see a screenful at a time, use the /P (Pause or Page) parameter:

```
C>TYPE myfile.txt /P
```

Press any key to see the next screen of text.

Renaming files (RENAME)

Sometimes you need to change the name of a file or group of files. To rename a file enter the command RENAME (you can

shorten it to `REN`), followed by the filename you want to change and then the name you want to change it to, for example:

```
C>RENAME myfile.txt newfile.txt
```

If the file you are renaming is not on the current drive, you need to state where it is:

```
C>RENAME a:myfile.txt a:newfile.txt
```

You can rename groups of files using the wildcard characters, for example:

```
C>REN *.bat *.old
```

Printing files

The operating system offers a number of utilities to facilitate the printing of files. The most significant is the print spooler, `CSPOOL`, which allows print requests to be queued and acted upon in an orderly fashion.

The most common method of printing is by way of a print command built into the application you are using. If you wish to print directly through the operating system, however, you can do so, using the `PRINT` command. Alternatively, you can copy a file to your logical print device, as follows:

```
C>COPY myfile.txt PRN:
```

For more information on printing, see the separate chapter on the subject, Chapter 13, “Printing.”

7.6 Keeping track of where you are

As mentioned earlier in this chapter, the system prompt is generally set up to indicate your current directory. So you always have a clear idea of where you are in the directory structure.

Changing your system prompt

You set the prompt with the `PROMPT` command. If you do not have a prompt which reflects your current directory, you can alter it to do so by typing:

```
C>PROMPT $P$G
```

The symbol \$P says show the current directory and the symbol \$G means show the greater-than sign (>).

If you entered this command and then changed to a directory called **sales** from the root directory you would see:

```
C:\SALES>
```

If you changed again to a subdirectory of **sales** called **foreign**, you would see:

```
C:\SALES\FOREIGN>
```

so you would always know what directory you are in.

You can put the PROMPT command into your **autoexec.bat** file, so that it is invoked every time you start up your computer. How to do this is described in Chapter 16, "Customizing your system."

Note

This is just one of the many ways in which you can change your prompt using the PROMPT command. See Chapter 11, "Commands," for more information.

Protecting your information

THE operating system offers a number of methods of protecting the information in your files. You can:

- ☐ Change your user password to prevent unwarranted access to your user account.
- ☐ Alter the access permissions for your files and directories, to ensure that you control who can read, write to or delete them.
- ☐ Temporarily disable your keyboard and screen.
- ☐ Password-protect individual files and directories.
- ☐ Set the attributes for individual files, to restrict their access or display.

The first three methods apply only to “secure systems,” systems, that is, in which user accounts have been established. The next two methods are less sophisticated, and apply to all systems regardless of whether or not user accounts are in place. If in any doubt, consult the person in charge of your system.

This chapter describes these various methods of protecting information. The chapter concludes with information on how to make backup copies of hard disks and diskettes.

8.1 Changing your user password

If your system has user accounts established, you will already have a user name and provisional password given you by the person in charge of the system, the “Superuser.”

To gain initial access to your system, you will have had to respond to a prompt such as:

```
Please enter your login name:
```

with your user name, then to a prompt such as:

```
Please enter your password:
```

with your provisional password.

You cannot change your user name (though the Superuser can). However, you can and indeed are expected to change your provisional password. Once you have done so, only you can gain entry to the system via your user name, and only you, as owner, control access to the files in your account.

You are also recommended to alter your user password every so often, to ensure that it remains a secret.

Valid user passwords

Valid passwords can be from three to 12 characters in length. As with filenames, the following characters are not permitted:

```
<> = , ; : . * ? [ ] ( ) / \ + ! ~
```

Passwords are not case sensitive; in other words, upper and lower case letters look the same to the operating system and so can be used indiscriminately.

Try to choose an unlikely password that no one can guess or decode. Foreign words and combinations of short strings are a good idea.

To change your user password, invoke the change password command, `PASSWORD`, without any parameters, as follows:

```
C>PASSWORD
```

A display such as the following appears:

```
Change login password for Ann
Please enter old password:
```

You should type your password as it currently stands. It will not show on your screen, as a security measure. You are now prompted:

```
Please enter new password:
```

You should now type the new password you have chosen. Again, what you type will not show on the screen.

To confirm your choice, you are finally asked:

Please confirm new password:

Retype your new password. Provided you have entered the same set of characters both times, having previously entered your old password correctly, your new password is accepted.

The next time you log in, you must give your new password when prompted.

Note

If you invoke `PASSWORD` with no parameters on a system that does not have user accounts installed, a different display appears, relating to file and directory passwording. This is described later in this chapter.

8.2 Changing Access Permissions

If user accounts have been established for your configuration, it follows that a system of access permissions for files and directories is also in place.

Access permissions allow different classes of user to perform different sorts of operation on the files and directories to which they relate.

Classes of user

For each file and directory the following classes of user exist:

- ☐ Owner (O)
- ☐ Group (G)
- ☐ World (W)

The owner of a file or directory is generally the user that created it, though ownership can be changed. A file or directory can only ever have one owner at any one time.

The group associated with a file or directory is the group to which its owner belongs. Each user account is set up to belong to a specific group. Like user names, group names cannot be changed by ordinary users. If necessary, the person in charge of

Types of access permission

your system can give you more information about who is in what group.

The world represents all the remaining users on the system; that is, all users who are not members of the owner's own group.

Each file can allow each class of user a different combination of access permissions. The following access permissions are available:

- ☐ Read (R)
- ☐ Write (W)
- ☐ Delete (D)

Read permission for a file means you can copy or display the file.

Write permission for a file means you can modify the file.

Delete permission for a file means you can rename or delete the file, or change its attributes.

Directory access permissions are, for each class of user, either total, allowing read, write and delete operations, or none at all.

When access permissions are displayed, they appear in the format:

```
rwd rwd rwd
```

where the three columns represent the permissions currently available for, from left to right, owner, group and world.

Where an access permission is not granted, a dash (-) appears in the appropriate location. A typical set of permissions for a file is, therefore:

```
rwd rwd r--
```

This means that the owner has read, write and delete permission, the group read, write and delete permission as well, and the world read permission only.

A typical set of permissions for a directory is:

```
rwd rwd ---
```

You can modify at any time the access permissions associated with existing files and directories in your ownership. Use the `XATTRIB` command to do this.

Default access permissions are a built-in part of each user account. You can change these default settings, and so change the access permissions automatically given new files on creation. Use the `XATTRIB /F` command to do this.

Using XATTRIB

The XATTRIB command lets you display and change access permissions.

To display permissions for a specific file, set of files or directory, use the format:

```
XATTRIB @wildspec
```

where *@wildspec* represents the file(s) or directory whose permissions you wish to inspect. (See Chapter 11, “Commands,” for a full definition of *@wildspec*.)

Omit the *@wildspec* parameter to display settings for all files in the current directory.

Here is an example of a possible XATTRIB display:

```
rwd --- --- fred      doc  C:\usr\fred\news.doc
rwd rwd --- joanne    doc  C:\usr\fred\joanne.txt
rwd --- --- fred      doc  C:\usr\fred\personal.doc
rwd rw- r-- fred      doc  C:\usr\fred\letter.doc
```

Note the three columns of access permissions on the left of the display.

To change access permissions, use the format:

```
XATTRIB [@wildspec] /class:permission...
```

For *class* type one or more of the letters O, G and W, for owner, group and world respectively. For *permission* type one or more of the letters R, W and D, for read, write and delete.

The existing settings remain in place for any class not specified in the *class* field. Where a permission is not specified, on the other hand, that permission is actively denied.

Repeat the */class:permission* combination for each different permission you wish to establish. Note that you cannot give greater access permissions to the world than to the group or owner. Similarly, you cannot give others in your group greater access permissions than you have yourself.

So, for example, to set group access permissions for the file **letter.doc** to read and write, but world permission to read only, you would type:

```
C>XATTRIB letter.doc /G:RW /W:R
```

To deny group and world users any access to **letter.doc**, you would type:

```
C>XATTRIB letter.doc /GW
```

A number of command switches are available with XATTRIB. One particularly powerful switch (/A) allows you to specify a new owner for a file or files. For example:

```
C>XATTRIB letter.doc /A:henry
```

assigns ownership of the file **letter.doc** to the user Henry.

For a description of all the possibilities of the XATTRIB command, see Chapter 11, "Commands."

Using XATTRIB /F

Default access permissions are established for each user when his or her user account is first set up. The XATTRIB command, when used with the /F switch, lets you display and change your default access permissions.

Changes to the defaults apply to the current working session only. To make such changes permanent, put XATTRIB /F settings in your **autoexec.bat** file.

To see what your current default access permissions are, type:

```
C>XATTRIB /F
```

Invoked in this way without any parameters, XATTRIB /F displays a message such as the following:

```
Default file create permissions for Ann: rwd r-- ---
```

Default permissions are shown for the three classes of user, owner, group and world.

To alter the default access permissions for the files and directories in your ownership, use the following format:

```
XATTRIB /F /class:permission...
```

For *class* type one or more of the letters O, G and W, for owner, group and world respectively. For *permission* type one or more of the letters R, W and D, for read, write and delete.

The existing default settings remain in place for any class not specified in the *class* field. Where a permission is not specified, on the other hand, that permission is actively denied.

Repeat the */class:permission* combination for each different default setting you wish to establish. So, for example, to set

group access permissions to read and write, but world to read only, you would type:

```
C>XATTRIB /F /G:RW /W:R
```

This, assuming the default for owner was previously set to read, write and delete, would result in the following message:

```
Default file create permissions for Ann: rwd rw- r--
```

Any file you subsequently created in the current working session would initially have the `rwd rw- r--` access permissions.

8.3 The LOCK command

You may not wish to log out every time you are away from your terminal for a short time. If you have processes running on any of your DOS sessions you will certainly not wish to do so. On the other hand, you may not wish to allow other users access to your terminal for the duration of your absence.

The operating sytem offers a solution to this dilemma. Using the LOCK command, you can disable your keyboard and screen. Once you have used LOCK, the screen clears and displays a message informing other users of its “locked” condition. For example:

```
This station is locked for user Ann  
To unlock please enter your login password:
```

No keyboard input, apart from password entry, is possible at this time; background processes on other DOS sessions, however, can continue to run without hindrance.

Only the user logged in at the terminal can reactivate keyboard and screen. To do so, simply type your user password.

8.4 Password-protecting files and directories

On systems that do not have user accounts established, you can still protect individual files and directories by associating them with their own passwords. Once established, the appropriate password must be entered on the command line for the file or directory to be accessed.

You can apply three levels of password protection to a file:

- ❑ The highest level prevents anyone from even reading the file unless they issue the password.
- ❑ The next level allows other users to read the file but not to change it without the password.
- ❑ The lowest level allows users to read and change the file, but not to rename or delete it without first giving the password.

You can also password-protect directories. If you do this, all the files and directories within the protected directory are completely inaccessible without the correct password.

You set passwords using the command `PASSWORD`. Passwords can be up to eight characters long. The `PASSWORD` syntax to set passwords on a system without user accounts is as follows:

```
PASSWORD [@]wildspec switch:password
```

PASSWORD switches

Three switches correspond to the levels of protection described above:

Switch	Meaning
/R	The highest level. /R sets the password attributes ---. You need the password to even read the file.
/W	The next level down. /W sets the password attributes r-- . You need the password to edit, copy, rename or delete the file, but not to read it.
/D	The lowest level. /D sets the password attributes rw-. You need the password to delete or rename the file or change the file's attributes.

Note

A full description of all the PASSWORD switches can be found in Chapter 11, "Commands."

Setting passwords

To set the password **lucky** with the highest level of protection to the file **wages.txt**, you type:

```
C>PASSWORD wages.txt /R:lucky
```

To set the password so that users can read and write to **wages.txt**, but not delete or rename it unless they issue the password, type:

```
C>PASSWORD wages.txt /D:lucky
```

Be careful when choosing a password. Make sure it is not too obvious to anyone attempting to work it out, or alternatively one that is too difficult to remember!

Issuing passwords

To issue a password after you have allocated one to a file, you type a semi-colon immediately after the filename followed by the password.

For example, the TYPE command is used to display a text file on the screen. To display the password-protected file **wages.txt**, you would type:

```
C>TYPE wages.txt;lucky
```

To copy the file **wages.txt** to a new file **archives.txt** on drive A:, enter the command:

```
C>COPY wages.txt;lucky a:archives.txt
```

Note that the copy, **archives.txt**, is not password-protected.

Password-protecting directories

In the same way you give files passwords, you can also allocate them to directories using other **PASSWORD** switches. To assign a password to a path, you use the password switch **/P:**. For example, to set a password on the directory **c:\clients\personal** type:

```
C>PASSWORD c:\clients\personal /P:private
```

To access this path now you have to issue the password **private** as follows:

```
C>CD c:\clients\personal;private
```

Global system passwords

With **PASSWORD** you can also set a universal or “global” password, that is, a password that the operating system automatically tries when you attempt to access a file or directory that is password-protected.

This saves you having to give the same password every time you try to access one of the files in the group.

This can be useful if you are working with an application that does not recognize passwords.

After you have issued the global password it is tried against each password-protected file you try to access. The global password remains in effect until it is switched off explicitly by you, or until the computer is restarted.

For example, if you had five files all protected with the password **leopard** and you wished to set the global password, you would type:

```
C>PASSWORD /G:leopard
```

Now you would not have to issue the password **leopard** for the files to which it applies.

Note

Remember to switch off the global password if you leave your computer unattended by entering the command **PASSWORD /NG**.

Removing passwords

From a file

To remove a password from a file, use the format:

```
PASSWORD filename /N
```

For example to remove the password **lucky** from the file **wages.txt**, enter:

```
C>PASSWORD wages.txt /N
```

You are prompted to give the password:

```
C:\Wages.txt ...file password?
```

Type in the password **lucky** (it is not shown on the screen) and press **Enter**. The password is then deleted, and the system prompt returned.

Alternatively, you can specify the password in the command line, for example:

```
C>PASSWORD wages.txt;lucky /N
```

From a directory

To remove a password from a directory or path, use the format:

```
PASSWORD dirpath;password /NP
```

For example to remove the password **private** from the path **a:\clients\personal**:

```
C>PASSWORD a:\clients\personal;private /NP
```

8.5 Setting file attributes

In addition to passwords, other pieces of information called *attributes* can be associated with a file. These attributes are as follows:

Attribute	Meaning
A	Archived. This attribute is set whenever a file is changed. It may be reset by, for example, BACKUP, FILELINK, REPLACE, XCOPY (and other commands), when a file has been “archived”.
H	Hidden. Setting up the hidden attribute makes the file invisible to the user. A hidden file will not appear in standard directory listings.
R	Read-only. When set, you can only read a file, you cannot write to it, or delete it.
S	System. Files with the system attribute set, are only displayed when the /S option is used with the DIR command.

Use the ATTRIB command to set and reset file attributes. When you set an attribute it means it is *active* or switched on. When you reset it, it is *non-active* or switched off.

To set an attribute, use the following format:

```
ATTRIB +attrib [@]wildspec
```

For example, to set the hidden attribute for all **.let** files in the DOC directory, you would type:

```
C>ATTRIB +H doc\*.let
```

To reset an attribute, use the following format:

```
ATTRIB -attrib [@]wildspec
```

Note

Attributes can provide a means of protecting your files.

This method, however, is not as safe as using passwords. For example, several commands have switches that allow you to alter files that have the read-only attribute.

8.6 Making backup copies of disks

It is extremely important that you make regular copies of your work to safeguard against loss of data. Make it a part of your working routine to take periodic backups of your files. Backups are copies of information taken from one disk and stored on another.

The operating system provides three commands for this purpose:

- ☐ For the hard disk, the BACKUP and RESTORE commands. These two commands always work together.
- ☐ For diskettes, the DISKCOPY command.

Backing up your hard disk

The BACKUP command enables you to copy all or part of a hard disk onto one or more diskettes. The original copy remains on the hard disk. You can then use your backup copies if the original is lost or corrupted, by restoring them using the RESTORE command. Any files backed up using the BACKUP command can only be used after you have restored them using RESTORE.

Establish a backup routine

When backing up your hard disk, you should try to establish a routine. Usually a backup of the entire hard disk is taken once every month or few months. Backing up the whole hard disk can be a lengthy task and requires a large number of diskettes. In between backing up the entire hard disk, you can back up any modified files using the various BACKUP command switches. How often you need to back up modified files depends on your working routine. You may need to backup files daily, or you may find once a week is enough.

Before you make any backups, keep in mind these important points:

- ☐ Do not backup disks affected by an ASSIGN, JOIN or SUBST command as they affect the way a disk is viewed by
-

the system; always cancel these commands first. See Chapter 11, "Commands."

- ❑ If your files and subdirectories have passwords, you need to give them first before taking a backup (see section 8.5).
- ❑ When you back up files onto a diskette, any files already on the diskette are overwritten. To avoid this happening there is a switch, /A, which tells the operating system to preserve the files already on the backup diskette.
- ❑ Make a note of the exact BACKUP command syntax you use to back up a disk, as you will need it when you restore your data, (see the section "Restoring BACKUP files").

Note

See Chapter 11, "Commands," for a description of the full range of switches available with the BACKUP command.

Backing up the entire hard disk

Before backing up a hard disk you need to make sure you have plenty of new, blank diskettes to hand. To backup a full 20 Mbyte hard disk you may need up to 35 formatted diskettes, though this number will vary according to the capacity of your diskette drive.

To backup your hard disk onto diskettes type:

```
C>BACKUP c: a: /S
```

The /S switch tells BACKUP to back up all subdirectories as well as the root directory (if you did not specify /S, only the root directory would be backed up).

You are asked to insert the first backup diskette into diskette drive A:

```
Backing up from C:\ to A:
Insert disk number 001 in drive A:
All files in the root directory of the
destination disk will be deleted
Strike a key when ready
```

The first diskette you insert is "backup diskette 001". Press any key to start the backup procedure.

When the first diskette is full, you are prompted to insert the next diskette (002). Remove the first diskette and label it with

its backup number (001), the date and the exact BACKUP command syntax you used. (It is important that you label your backup diskettes in the sequence they are created; this is due to the way in which the RESTORE command works. Also you will need to use the same syntax you used to backup your files to restore them.)

Insert the next diskette, 002, and press any key to continue. When this diskette is full, remove it, label it, and then insert diskette 003. Continue doing this until the system prompt is returned. This may take some time depending upon the size of your hard disk.

You now have a full set of backup diskettes for your hard disk. These you should write-protect forthwith, using write-protect tabs (see Chapter 5, "Getting started").

Backing up modified files

To backup files changed since your last backup was taken, you use the switch /M (M means modified) as follows:

```
C>BACKUP c: a: /M
```

If you intend to backup your modified files onto an existing backup diskette, you must specify the switch /A in the command line:

```
C>BACKUP c: a: /M /A
```

This is because backing up onto diskettes erases all the files that are already stored there. The /A switch tells BACKUP to preserve the files that already exist. You can also backup files modified since a specified date onto your backup diskettes using another switch, /D:date. For example:

```
C>BACKUP c: a: /D:8-3-90
```

backs up all files changed since March 8th 1990. You can only separate the items in the date with hyphens. The date format you use depends on what country your system is set up for.

Backing up directories

To backup a specified directory and all its associated subdirectories, use the /S switch in the same way you used it to backup the root directory and all its associated directories. For

example to back up the directory **c:\accounts** and all its subdirectories, type:

```
C>BACKUP c:\accounts a: /S
```

To back up a single file, specify the filename in the command line. For example:

```
C>BACKUP c:\accounts\report.txt a:
```

backs up the file **report.txt** onto a diskette in drive A:.

Using wildcard characters

You can use wildcards in the command line. For example:

```
C>BACKUP *.doc a:
```

backs up all files in the current directory with the extension **.doc** onto a diskette in drive A:.

Restoring backup files

The **RESTORE** command is always used to recover files stored using **BACKUP**. It works in a similar way to **BACKUP**. It copies backup files back into a usable form, usually onto the disk from which they originally came. If you labeled your diskettes correctly when you backed them up, you just insert them in the correct order. You need to use the equivalent syntax you used when backing up your files to retrieve them, so if you want to restore your complete hard disk backup type:

```
C>RESTORE a: c: /S
```

(The corresponding **BACKUP** command is **BACKUP c: a: /S**.)

You are prompted to insert the first **RESTORE** diskette, which would be the first diskette you backed up onto:

```
Insert disk number 001 in drive A:  
Strike a key when ready
```

Press any key to continue. **RESTORE** displays the name of each file as it restores it to the disk. You are prompted as above until all the diskettes are restored, then the system prompt is returned.

To restore the directory **accounts** and its associated directories type:

```
C>RESTORE a:\accounts c: /S
```

The RESTORE command has many switches which are similar to those of the BACKUP command. See Chapter 11, “Commands,” for a description of the full range of switches available with the RESTORE command.

Checking what is on a backup disk

The RESTORE command has a helpful switch /R which reports on what files would be restored without taking any action. This is very useful for checking what files are on your backup diskettes.

Copying diskettes

The DISKCOPY command duplicates entire diskettes. The basic format of the command for single diskette drive systems is:

```
C>DISKCOPY a: a:
```

For twin diskette systems type:

```
C>DISKCOPY a: b:
```

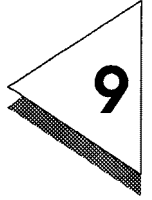
In either case, follow the screen prompts. DISKCOPY will automatically format your diskette for you if it is not already formatted. For more information, see Chapter 11, “Commands.”

Caution

DISKCOPY overwrites anything that exists already on the destination diskette, that is, on the diskette you are copying to. If you wish to add files to a diskette without necessarily overwriting existing data, use the COPY or XCOPY commands.

Always write-protect diskettes containing important information.

The EDITOR



THIS chapter describes how to use EDITOR, a full-screen editor which you can use to create and edit text files. You can use it with any keyboard, and there are shorthand forms for some of the common commands. You can use EDITOR to:

- ☐ Create a new file.
- ☐ Browse through a file.
- ☐ Edit a file.

9.1 Starting EDITOR

There are two ways you can start EDITOR from the command prompt:

- 1 Type EDITOR followed by the name of the file you want to edit or create and press **Enter**, for example:

```
C>EDITOR myfile.txt
```

When you start EDITOR it looks for the file you specified (in the above example, **myfile.txt**) on the current drive, unless you specify a different drive or pathname before the name of the file.

If EDITOR finds the file, it displays it on screen. If EDITOR cannot find the file it displays a message such as:

```
c:\myfile.txt . . . file not found
Create new file (Y/N)?
```

Type Y to create the file, or N to go to the EDITOR title screen where you can type in another filename, or press **Esc** to leave EDITOR.

- 2 Type EDITOR and press **Enter**. This takes you directly to the EDITOR title screen.

Type in a filename and press **Enter**, or press **Esc** to exit from the title screen back to the command prompt.

9.2 Quitting EDITOR

You can quit editing a file in three ways:

- ☐ To save your edited file and exit directly, press **Ctrl+KX**.
- ☐ To save your edited file and return to the EDITOR title screen, press **Ctrl+KD**.
- ☐ To abandon any changes made since the last time the file was saved and return to the EDITOR title screen, press **Ctrl+KQ**.

EDITOR asks you if you really want to quit without saving your file, unless you have not made any changes. You are then returned to the EDITOR title screen, where you can

press **Esc** to quit EDITOR. **Ctrl+KQ** is useful if you want to abandon a file that you opened by mistake.

Note

You can press **Ctrl+R** at the EDITOR title screen to display the previous filename edited, and press **Enter** to go back to editing that file.

9.3 Entering text

To enter text, type as you would if using a word processor or typewriter except, when you reach the end of a line, press **Enter** to start a new line. This is because in EDITOR there is no “word wrap”; it does not automatically start a new line unless you tell it to.

Try entering the following text, remembering to press **Enter** at the end of each line:

```
REM this batch file changes to the Wundawrd directory
REM and starts my word processor
cd \wundawrd
ww
```

This text creates a simple batch file used to start a word processor called Wundawrd. A batch file is a quick way of carrying out a series of frequently-used commands, you simply type the batch file name at the command line and all the commands in the file are carried out for you; see the following chapter for more information. REM is a batch file subcommand.

Notice as you type, at the top of the screen EDITOR keeps track of how many characters and columns you are entering:

```
Chr=108 Col=3
```

It tells you exactly which character and column the cursor is on. Also, your directory path and filename are displayed at the top of the screen.

Using the Tab key

You can use the Tab key to move the cursor across the page and insert text at set points. The tab stops are preset to every eighth column across the screen. You can think of the **Tab** key as meaning “add space characters until you reach the next tab stop”.

Moving text down a line

If you press **Ctrl+N** when editing text, the rest of the line at the cursor is taken down to the next line, while the cursor remains where it is. For example, if the cursor was positioned after “starts” in this line:

```
REM and starts my batch file
```

and you press **Ctrl+N**, “my batch file” would move to the next line, like this:

```
REM and starts  
my batch file
```

while the cursor is still positioned after “starts”.

Entering control characters

You may wish to put control characters in a file, for example to control a printer. You cannot do this directly as EDITOR interprets control characters as commands.

However, if you precede the control character by an additional **Ctrl+P**, EDITOR understands that your entry is in fact to be placed in the file you are editing.

For example, to insert **Ctrl+A** into a file, you would press **Ctrl+P** followed by **Ctrl+A**.

Getting online help

You can get help in EDITOR by pressing **F1** or **Ctrl+J**. To exit from the help screen, press **Esc**.

9.4 Moving the cursor

You can move the cursor around your file in steps of one character, one word or one page, or you can jump directly to the start or end of the file.

To move the cursor one character

- ☐ To the right, $\rightarrow$ or **Ctrl+D**.
- ☐ To the left, $\leftarrow$ or **Ctrl+S**.
- ☐ Up one line, $\uparrow$ or **Ctrl+E**.
- ☐ Down one line, $\downarrow$ or **Ctrl+X**.

Note

On some keyboards the four keys **D**, **S**, **E** and **X** form an approximate diamond shape on your keyboard, sometimes called the “cursor-control diamond”. The position of these keys may help you to remember what they do.

To move the cursor one word

- ☐ To the left, **Ctrl+A**.
- ☐ To the right, **Ctrl+F**.

The keys **A** and **F** are also part of the “cursor-control diamond”.

To move the cursor one page

- ☐ To the previous page, **PgUp** or **Ctrl+R**.
- ☐ To the next page, **PgDn** or **Ctrl+C**.

A “page” is the number of lines displayed on your screen, less eleven. On a 25-line display, pressing **PgDn** moves the cursor fourteen lines down. Some of the previous page will still be displayed on the screen, to make it easier for you to find your place again.

To move the cursor to the start or end of a line

- ❑ To the beginning of a line, press **Ctrl+ QS**.
- ❑ To the end of a line, press **Ctrl+ QD**.

To move the cursor to the start or end of a file

- ❑ To the beginning of a file, press **Home** or **Ctrl+QR**.
- ❑ To the end of a file, press **End** or **Ctrl+QC**.

The cursor-control diamond

To retrace our steps, here are all the cursor movement control keys. Again note the arrangement of command letters into the “cursor-control diamond” to help you remember these commands.

Start of file

Home (Ctrl+QR)

Previous line

↑ (**Ctrl+E**)

Previous page

PgUp (Ctrl+R)

Start of line	Word left	Char left	Char right	Word right	End of line
Ctrl+QS	Ctrl+A	← (Ctrl+S)	→ (Ctrl+D)	Ctrl+F	Ctrl+QD

Next line

↓ (**Ctrl+X**)

Next page

PgDn (Ctrl+C)

End of file

End (Ctrl+QC)

9.5 Inserting and overwriting text

EDITOR has two modes for entering text: insert (ins) and overwrite. When you start EDITOR, it is automatically in insert mode. The message `ins` is shown at the top rightmost corner of the screen.

To insert text, position the cursor at the point where the new text is to start and type it in. Existing text moves to the right to make room for what you type.

You can switch to overwrite mode by pressing **Ins** or **Ctrl+V**. Now when you type you will overwrite whatever the cursor is positioned on.

To switch overwrite mode off, press **Ins** or **Ctrl+V** again.

Note

Remember if `ins` is at the top of the screen, you are in insert mode. If `ins` is not displayed, you are in overwrite mode.

Inserting a file at the cursor

To read in a file, place the cursor where you want the file to be inserted, and press **Ctrl+KR**. You are prompted:

```
Name of file to read in?
```

Type in the name of the file you want to insert and press **Enter**. The file is inserted at the cursor. The remaining text is moved down to accommodate the new text.

9.6 Deleting text

Text can be deleted a character at a time, a word at a time or one line at a time. (You can also delete blocks of text; see the next section, “Working with blocks of text”.)

Deleting a character

- ❑ To delete the character the cursor is on, press the **Del** key or **Ctrl+G**.

The character is deleted from the screen, and the text to the right of the cursor shifts to the left to fill the gap.

If you press **Del** or **Ctrl+G** repeatedly, characters will continue to be deleted until you reach the end of the file.

- ❑ To delete the character in front of the cursor, press the **Backspace** key or **Ctrl+H**.

The character is deleted from the screen and the text to the right of the cursor shifts to the left to fill the gap.

If you press **Backspace** or **Ctrl+H** repeatedly, characters continue to be deleted until you reach the beginning of the file.

For example:

If you typed Wundaief instead of Wundawrd, place the cursor on i and press the **Del** key until the letters i, e, and f, are deleted. The line containing the mistyped Wundawrd would change like this, with the cursor positioned at the end of the line:

```
cd \Wunda
```

Now you type in the correct characters, that is wrd.

If you press **Del** at the end of a line, you delete the invisible “newline” character and bring up the next line. Similarly, pressing ← (**Backspace**) or **Ctrl+H** at the beginning of a line moves that line up to the end of the previous line.

Note

If you want to delete more than one character, it is usually quicker to use one of the key combinations described in the following sections.

Deleting a word

In EDITOR a “word” is defined as a string of characters that starts at the current cursor position and continues up to and including the first following space.

- ❑ To delete a whole word, position the cursor at the beginning of the word to be deleted and press **Ctrl+T**.
- ❑ To delete part of a word, place the cursor within the word and press **Ctrl+T**. The letters at and beyond the cursor are deleted. The text to the right of the cursor shifts to the left to fill the gap.

For example:

If the cursor is positioned at the start of the word “changes” in the following line:

```
REM this batch file changes to the Wundawrd directory
```

pressing **Ctrl+T** repeatedly changes the appearance of the line like this:

```
REM this batch file to the Wundawrd directory
REM this batch file the Wundawrd directory
REM this batch file Wundawrd directory
REM this batch file directory
REM this batch file
```

Deleting a line

Position the cursor anywhere in the line to be deleted and press **Ctrl+Y**. The line is deleted and lines below it move up to fill the gap.

For example:

If the cursor was positioned in the second line of the following text:

```
REM this batch file changes to the Wundawrd directory
REM and starts my word processor
cd \wundawrd
ww
```

Pressing **Ctrl+Y** changes the appearance of the display like this:

```
REM this batch file changes to the Wundawrd directory
cd \wundawrd
ww
```

Pressing **Ctrl+Y** again produces this result:

```
REM this batch file changes to the Wundawrd directory
ww
```

Pressing **Ctrl+Y** once more, produces this display:

```
REM this batch file changes to the Wundawrd directory
```

Note

On some keyboards, the keys **G**, **H**, **T** and **Y** (for deleting characters, words and lines) are next to each other and form an approximate square. You may find this helps you to remember these commands.

9.7 Working with blocks of text

You can mark a section or block of text and then move, delete, copy it, or store it into a separate file. A block of text can be any size (according to how much disk space you have got).

Marking a block of text

To mark a block of text, place the cursor at the start of the section you want to mark and press **Ctrl+KB**. The text is marked with . Move the cursor to the end of the block of text and press **Ctrl+KK**. The end of the text block is marked with <K> and the text block is highlighted.

Moving a block of marked text

To move a block of marked text, place the cursor where you want the block to be moved to and press **Ctrl+KV**. The text is erased from its original position and moved to where the cursor is. The text moves up to fill the gap.

Deleting a block of marked text

To delete a block of marked text, press **Ctrl+KY**. The text is deleted and the text beneath it moves up to fill the gap.

Copying a block of marked text

To copy a block of marked text, move the cursor to where you want the block of text copied to, press **Ctrl+KC**. The block of text is copied to the cursor position, still marked. The original text remains where it is and is no longer marked.

Storing a block of marked text in a file

To store a block of marked text in a separate file, press **Ctrl+KW**. You are prompted:

Name of file to write to?

Type in the name of the file where the text is to be stored and press **Enter**. The file is created and the block of marked text is written to it. If the file already exists, you are asked:

File already exists, overwrite (Y/N)?

Type Y to overwrite the file, N if you want to keep the file as it is. If you answer N no action is taken; press **Ctrl+KW** again and enter a different filename.

9.8 Saving your work

You can save files in three ways:

- ☐ To save and exit from the EDITOR, press **Ctrl+KX**.
- ☐ To save and continue editing your file, press **Ctrl+KS**.
- ☐ To save your work and begin editing a new file, press **Ctrl+KD**.

This saves your work and updates your backup file, as described in the following section.

9.9 Automatic backup files

When you open an existing file, a “working copy” of the file is stored in memory. The original file remains on your disk, unaltered.

When you save your edited file, the working copy is stored on the disk and is given the name of the original file. The old file becomes an EDITOR backup file with the extension **.bak**.

For example:

If you open the file **mytext.doc** and edit it, when you save the file the following happens:

- ☐ The original file **mytext.doc** is renamed to **mytext.bak**.
- ☐ The new version is stored with the name **mytext.doc**.

You can use backup files to recover files. For example if the current version is:

- ☐ found to be incorrect.
- ☐ accidentally erased.

If you accidentally deleted the file **mytext.doc**, you could revert to the **.bak** file by typing:

```
C>COPY mytext.bak mytext.doc
```

Note

To prevent any filename conflicts, EDITOR does not allow you to edit files with the extension **.bak**.

9.10 Technical details

- ❑ EDITOR produces ASCII files with no embedded specific control characters, so its output can be used for batch and similar files. It allows 8-bit characters to be entered and displayed.
- ❑ There is no size limit on the files you can edit, apart from the size limit imposed by the amount of disk space you have available.
- ❑ The **Enter** key inserts two characters into the file being edited: a carriage return (ASCII code 13) and a line feed (ASCII code 10), in that order. Note that a single **CR**, a single **LF** or a **CR/LF** pair is always treated as a single character when moving through the text.
- ❑ The **Tab** key inserts a character (ASCII code 09) into the file. On screen the **Tab** character inserts spaces from the cursor position to the next tab stop. A tab stop is preset at every eighth column across the screen.
- ❑ EDITOR does not insert any special End-of-File marker, but will recognize the End-of-File marker code 26 (ASCII **Ctrl+Z**) used by some older style editors.
- ❑ EDITOR provides automatic creation of **.bak** backup files. It does not allow you to edit files with a **.bak** extension.
- ❑ During use, EDITOR creates two temporary files in the same directory as the file you are editing. These are automatically deleted when you quit EDITOR.

9.11 Command summary

Table 9-1 Summary of EDITOR Commands

Keys	Function
Getting help	
F1 or Ctrl+J	Displays help text
Moving the cursor	
Home or Ctrl+QR	Start of the file
End or Ctrl+QC	End of the file
Ctrl+QS	Start of the line
Ctrl+QD	End of the line
← or Ctrl+S	Character left
Ctrl+A	Word left
→ or Ctrl+D	Character right
Ctrl+F	Word right
↑ or Ctrl+E	Previous line
↓ or Ctrl+X	Next line
Ctrl+W	Scroll text down one line
Ctrl+Z	Scroll text up one line
PgUp or Ctrl+R	Previous page
PgDn or Ctrl+C	Next page
Deleting text	
Del or Ctrl+G	Delete character
Ctrl+Y	Delete line
Backspace or Ctrl+H	Delete character to the left
Ctrl+T	Delete word to the right

Table 9-1 Summary of EDITOR Commands (continued)

Keys	Function
Working with blocks of text	
Ctrl+KB	Mark the start of a block of text
Ctrl+KK	Mark the end of a block of text
Ctrl+KV	Move a marked block of text
Ctrl+KY	Delete a marked block of text
Ctrl+KC	Copy a marked block of text
Ctrl+KW	Store a marked block of text in a file
Saving files	
Ctrl+KX	Save changes, exit
Ctrl+KD	Save changes, edit new file
Ctrl+KS	Save changes, resume
Ctrl+KQ	Abandon changes, edit new file
Other commands	
Ins or Ctrl+V	Switch between insert and overwrite modes
Ctrl+KR	Read in a file at the cursor
Enter	Insert a new line at the cursor
Ctrl+N	Insert a new line after the cursor
Ctrl+P	Insert a control character at the cursor

Batch processing

USUALLY when you work with your computer you type in commands at the command line and the operating system carries them out. This is interactive processing. There is another kind of processing you can use called batch processing.

With batch processing you collect a series of commands together into a batch file. After you type the name of the batch file at the command line, each command in the file is executed in turn, just as if you had typed the commands one at a time. Batch files can have any eight character name, but they must always have the extension **.bat**.

This chapter serves as an introduction to batch files. For examples of batch files that facilitate the running of applications, see Chapter 15, "Running applications."

10.1 Using batch files

As you work regularly with your system you will probably find yourself typing the same command or sequence of commands over and over. For example, you may have a set procedure each morning to load your spreadsheet program, and another procedure in the evening to backup all the files you created or altered that day, or you might regularly print a group of files.

In batch processing you can group all the commands you need to carry out these tasks into one file to save yourself the bother of typing them all in.

For example, if you used the following commands to tidy your disk and backup your files at the end of each day:

```
C>CD \letters
```

```
C>ERASE *.old
```

```
C>MD a:\dayswork
```

```
C>COPY *.doc a:\dayswork
```

you could create a batch file to contain all these commands called **dayover.bat**. Then at the end of each day, all you would need to type would be:

```
C>dayover
```

In this example, the operating system first carries out the command `CD \letters` (change to the subdirectory **letters**). It then returns to the batch file and carries out the command `ERASE *.old` (delete all files with the extension ***.old**). Next it processes the command `MD a:\dayswork` (make the subdirectory **dayswork** on drive A:). Finally it returns to the batch file to carry out the command `COPY *.doc a:\dayswork` (copy all files with the extension **.doc** to the subdirectory **dayswork** on a diskette in drive A:).

Note that each new command in the batch file must start on a new line.

Batch files are a flexible and powerful feature of your system. You can create very simple to very complex batch files. They are useful for shortening commands, and for ensuring a procedure is carried out the exact same way every time.

10.2 Creating batch files

You can create batch files with EDITOR, a word processor or with the COPY command (see Chapter 11, “Commands”). Below is an example of how to use EDITOR to create a simple batch file. (If you create the file with your favorite word processor, make sure the finished file is in plain text, that is ASCII format; refer to your word processor manual for details.)

To create the example file **dayover.bat**:

- 1 From the command prompt type:

EDITOR dayover.bat

As this is a new file you see a message similar to this:

```
C:\dayover.bat ....file not found
Create new file (Y/N)?
```

Type Y for Yes and the file is created; the EDITOR screen is displayed.

- 2 Type the commands you want to enter in the batch file, pressing **Enter** at the end of each line:

```
CD \letters
ERASE *.old
MD a:\dayswork
COPY *.doc a:\dayswork
```

- 3 Press **Ctrl+KX** to save the file and quit EDITOR.

A similar procedure can be used to create any batch file.

10.3 Running batch files

Any batch file that you want to run must be located in either the current directory or one of the directories in the search path set up by the PATH command. (The PATH command is described in Chapter 11, “Commands”).

To run the batch file, simply type the batch file’s filename. You do not need to type the **.bat** extension.

So, to run the file **dayover.bat**, all you need type is:

```
C>dayover
```

The system prompt returns when the batch file is finished.

Stopping a batch file

To stop a batch file while it is running, press **Ctrl+C** and you will see:

```
Halt Batch Process (Y/N)?
```

Type Y for Yes and the remaining commands in the batch file are *not* carried out; you are returned to the system prompt.

Type N for No and the remaining commands are processed.

If **Ctrl+C** fails to interrupt the batch file, you can press **Ctrl+Alt+Del** to interrupt the DOS session (provided global **Ctrl+Alt+Del** is disabled).

10.4 Making general purpose batch files

You can create general purpose batch files, that is, batch files that can be made to do different things by adding parameters when they are run.

To allow this kind of thing to happen you need to use replacement variables in your **.bat** file. Replacement variables take the form *%n*, where *n* is a number from zero through nine.

When the batch file is run, the variables it contains are replaced by the different strings found on the command line.

The variable *%0* represents the name of the batch file and its drive specification, if included. *%1* represents the first parameter on the command line, *%2* stands for any second parameter, *%3* the third and so on. Use the SHIFT subcommand to alter this correlation.

For example, suppose you often moved files from the subdirectory **c:\amanda\letters** to another subdirectory called **c:\queries\donewith**. You would need a general-purpose batch file that copied *any* file from the first subdirectory to the second, and then deleted the original file.

So, you could create a batch file called **moveover.bat**, in the **c:\amanda\letters** directory, containing the following lines:

```
COPY %1 c:\queries\donewith  
ERASE %1
```

Then, if you typed the batch file command:

```
C>MOVEOVER jones.let
```

the commands actually carried out would be:

```
C>COPY jones.let c:\queries\donewith
```

followed by:

```
C>ERASE jones.let
```

If you typed the batch file command:

```
C>MOVEOVER smith.let
```

the commands carried out would be:

```
C>COPY smith.let c:\queries\donewith
```

followed by:

```
C>ERASE smith.let
```

Remember these points when including variables in your general purpose batch files:

- ☐ You can specify up to 10 variables (%0 through %9) at any one time. If you need to use more than 10 variables, see the SHIFT command.
 - ☐ Always separate variable values by a space, and make sure that values are entered in the proper sequence.
 - ☐ If you want to use % as a filename within a batch file, you must include it twice. For example, if the filename is **test%.dat**, the batch file must specify **test%%.dat**. It is generally not good practise to use the percentage sign in filenames.
-

Environment variables

Batch files can also retrieve and use environment variables such as “OS” (defining the operating system), “VER” (the current version number), and “path” (the current path). See also the SET command in Chapter 11, “Commands.”

The environment variable names must be surrounded by % signs when used in batch files.

For example, including the following command in the **autoexec.bat** file:

```
PROMPT %OS% $P$G
```

changes your default prompt, so that it displays the name of the operating system in front of the current directory and > symbol.

10.5 Batch file subcommands

Batch files can be made to be much more than a simple list of operating system commands and variables. Using batch file subcommands you can create batch files that will:

- ☐ Call a second batch file from within the first one.
- ☐ Display (or not display) the batch file commands on the screen.
- ☐ Prompt for keyboard input.
- ☐ Facilitate the construction of menus of choices.
- ☐ Cause the batch file to pause while you change diskettes.
- ☐ Display a message on the screen.

Other batch file subcommands enable you to write batch files that are like simple programs. Such batch files can take decisions and process files in different ways depending on the type of file.

Note

Batch processing subcommands are all internal commands.

Table 10-1 contains an alphabetical list of all the batch processing subcommands.

Table 10-1 Batch Processing Subcommands

Command	Explanation
@command	Prevents a command from being displayed.
:label	Defines a label which can be referenced by a GOTO batch subcommand.
CALL	Calls a batch file from within a batch file.
ECHO	Allows or inhibits the display of command lines, or displays messages.
ECHOERR	Allows or inhibits the display of command lines, or displays messages, regardless of any redirection that may be in place.
FOR	Allows repetitive execution of a command for each file specified.
GOTO	Transfers control to the line following a label.
IF	Provides conditional execution of commands.
INPUT	Accepts keyboard input (terminated by a carriage return) and assigns it to an environment variable.
INPUTC	Accepts a single character at the keyboard (without an accompanying carriage return) and assigns it to an environment variable.
PAUSE	Stops batch file processing until the user presses any key.
PAUSEERR	Stops batch file processing until the user presses any key. Displays a message, regardless of any redirection that may be in place.
REM	Allows comments to be inserted into a batch file.
SHIFT	Makes it possible to change the position of command line parameters to access more than 10 (%0 through %9) replacement variables.

Command conventions

The following conventions are used in the command descriptions:

[]	Square brackets indicate an optional part of the command line. Do not type the square brackets themselves, for example for [/C] you type /C.
	A vertical bar indicates an either/or choice, for example ON OFF. Choose just one of the options, and do not type the vertical bar.
command	Indicates operating system command.
filespec	Represents a file specification which can include some or all of the following: [d:] [dirpath] filename [.ext] [;password] where <i>d:</i> is the drive (optional), for example C: <i>dirpath</i> is the location of the file in the directory structure, for example \work\ or \. <i>filename</i> is the 1-8 character name of the file. <i>[.ext]</i> is the optional 1-3 character file extension, for example .bat. <i>[;password]</i> is the optional 1-8 character password.
n	Represents a number you type in. The values you can substitute are given in the command explanation.

@command

Format

@command

Explanation

If you place the @ symbol in front of a command, it will prevent it from being displayed when a batch file is executed. For example if you place @ in front of the command ECHO OFF, ECHO OFF will not be displayed.

:label

Format

:label

Explanation

:label defines a label which can be referenced by a GOTO batch subcommand. Only the first eight characters after the colon are considered part of the label, additional characters are ignored. Labels are not displayed during batch file execution.

CALL

Format

CALL filespec

Explanation

CALL allows a batch file to load and run a second batch file, and then returns to execute the remaining commands in the original batch file; the PATH is searched when the second batch file is looked for.

CALL subcommands can be placed at any line in the batch file.

Example

The following example calls the batch programs **second.bat** and **third.bat** and processes them in turn:

```
CALL second.bat
```

```
CALL third.bat
```

ECHO

Format

ECHO [ON|OFF|message]

Explanation

ECHO allows or inhibits the screen display of commands executed from a batch file. It does not affect the messages produced during command execution.

ECHO is ON by default, and shows all commands as they run. ECHO OFF stops all commands from appearing on the screen.

The ECHO message option displays messages on the screen whether or not ECHO is ON or OFF. You can display several lines of text in this way:

```
ECHO This is my very  
ECHO own batch file
```

The current status of ECHO is displayed if ECHO is issued from the command line with no parameters.

Note

Preceding a command with the @ character prevents the command from being displayed. For example, if a batch file included the command @ECHO OFF, the words `echo off` would not be displayed on the screen when the batch file is run.

@ECHO OFF is often used at the start of a batch file to prevent all commands from being displayed.

ECHOERR

Format

ECHOERR [ON|OFF|message]

Explanation

ECHOERR allows or inhibits the screen display of commands executed from a batch file. It does not affect the messages produced during command execution.

ECHOERR is ON by default, and shows all commands as they run. ECHOERR OFF stops all commands from appearing on the screen.

The ECHOERR message option sends messages to the standard error device (the screen), whether ECHOERR is ON or OFF and regardless of whether any redirection is in operation. Its principal use is, therefore, to send error messages to the screen, when all remaining batch file output is being redirected somewhere else.

You can display several lines of text in this way:

```
ECHOERR This is my very  
ECHOERR own batch file
```

The current status of ECHOERR is displayed if ECHOERR is issued from the command line with no parameters.

Note

Preceding a command with the @ character prevents the command from being displayed. For example, if a batch file included the command @ECHOERR OFF, the words `echoerr off` would not be displayed on the screen when the batch file is run.

@ECHOERR OFF is often used at the start of a batch file to prevent all commands from being displayed.

FOR

Format

FOR %%variable IN (fileset) DO command

Explanation

FOR allows for repetitive execution of a command. Each filename in the set is sequentially substituted in the command following DO. Wildcards are valid for members of the set and all matching filenames on the disk are substituted in the command. Only one FOR command can be used on a command line.

Example

If you enter this command in a batch file:

```
FOR %%f IN (myprog.asm acct.bas acct2.bas) DO type %%f
```

the three specified files are each typed to the screen in turn. If you enter a command like this at the command line, you only need enter one set of % signs, for example:

```
FOR %f IN (myprog.asm acct.bas acct2.bas) DO type %f
```

GOTO

Format

GOTO label

Explanation

The GOTO subcommand means processing of the batch file will continue from the line following on from the label specified. If the label is not found, the current batch file ends with this message:

```
Label ... not found
```

To set up a label in a batch file, use a colon (:) followed by a character string. The first eight characters of the string are considered the significant characters, and they must be different from those of all other labels.

Labels within a batch file are not shown while the file is executing. Labels not referenced by a GOTO subcommand may be used to place comments in your batch file.

Example

The following batch file displays the file **info.txt** continuously, until you press **Ctrl+C** or **Ctrl+Break**:

```
:start  
TYPE info.txt  
GOTO start
```

IF

Format

IF [NOT] condition [AND|OR [NOT] condition]... command

Explanation

IF allows conditional processing of operating system commands. The statement is executed when the IF condition is true. When the condition is false, the command is ignored and the next statement is executed.

If you include the NOT parameter, the system will carry out the command when the condition is false.

condition can be one of the following:

☐ `string1 == string2`

When *string1* and *string2* are identical, this condition is true.

☐ `KEY [remark] == character`

True if *character* is input.

☐ `ERRORLEVEL number`

This condition is true if the previous program had an exit code of greater than or equal to *number*. Your programs can set an error code that can then be checked by the IF ERRORLEVEL command.

☐ `EXIST filespec`

If *filespec* is on the indicated drive, EXIST *filespec* is true.

Pathnames are allowed with the filespec.

☐ `GROUPNAME == string`

True if *string* equals the name of the group in question.

Available on secure systems only.

☐ `USERID == nnnn`

True if *nnnn* is the four-character hex id of the user in question.

Available on secure systems only.

☐ `USERNAME == string`

True if *string* equals the name of the user in question. Available on secure systems only.

Note

Many operating system commands set the error level, which can then be tested with an `IF ERRORLEVEL` command; see Chapter 11, “Commands,” for details.

Examples

The following example illustrates the `IF EXIST` form:

```
IF EXIST myfile GOTO xyz
DIR B:
.
.
:xyz
command
```

This condition is true if executed with **myfile** on the default drive. The *command* following the label **xyz** is executed. If **myfile** is not found, the `GOTO xyz` is not executed and processing continues with the `DIR B:` command.

The following example illustrates the `IF string1==string2` command:

```
IF "%1"=="John" ECHO What are you doing here?
```

The batch file with this command, assuming **John** is named as the `%1` parameter, performs the `ECHO` batch command. What are you doing here? is displayed. If **John** is not given as the `%1` parameter, the condition is false, and the `ECHO` batch command is not executed.

Note

You should always place quotation marks around the text string, as this prevents syntax errors if no parameters are specified.

The following example illustrates the `IF ERRORLEVEL` form:

```
XCOPY %1 c:\queries\donewith  
IF ERRORLEVEL 1 GOTO nodelete  
DEL %1  
:nodelete
```

The above example further enhances the batch file **moveover.bat** (described in section 10.4) to only delete a file if the file is copied first. The `XCOPY` command is used to copy the file. When it completes, `XCOPY` sets the errorlevel to 0 if it finishes successfully, and to 1 if it fails. If it fails, the condition is true and the line `IF ERRORLEVEL 1 GOTO nodelete` is processed. This ensures the file is not deleted unless it has been copied first.

The following example illustrates the `IF NOT EXIST` form:

```
IF NOT EXIST a:%1 COPY b:%1 a:  
myprog1
```

This batch file is looking for a certain file to be on drive A:. The `IF` condition is executed before **myprog1** to ensure that the required file is on drive A:. If the file is not on drive A:, the condition is true and the copy command is processed, copying the file from drive B: to drive A:. If the file is on drive A:, the copy is not processed.

INPUT

Format

```
INPUT ["message"] env
```

Explanation

Accepts keyboard input (terminated by a carriage return) and assigns it to the environment variable *env*. *Message* is an optional prompt for input.

Example

The following example prompts for keyboard input, and assigns it to the environment variable **test**:

```
INPUT "Enter a sample value" test
```

INPUTC

Format

```
INPUTC ["message"] env
```

Explanation

Accepts a single character at the keyboard (without an accompanying carriage return) and assigns it to the environment variable *env*. *Message* is an optional prompt for input.

Example

The following example prompts for keyboard input, and assigns it to the environment variable **test2**:

```
INPUTC "Type any character" test2
```

PAUSE

Format

PAUSE [remark]

Explanation

You can insert a PAUSE subcommand into a batch file to delay the processing of the file. The pause in processing allows you to let an action occur, such as changing disks.

PAUSE always shows the following message:

Strike a key when ready . . .

The optional remark can contain a maximum of 122 characters.

Example

The following example shows a PAUSE subcommand in a batch file:

```
COPY a:*. * b:
PAUSE Insert new disk in drive A
COPY a:*. * b:
```

When the batch file is run, the first diskette in drive A: is copied. Then processing is suspended and this message appears:

```
PAUSE Insert new disk in drive A
Strike a key when ready . . .
```

Two responses are possible:

- ☐ Insert the new diskette and press any key to continue.
- ☐ Press **Ctrl+C** to stop the processing.

PAUSEERR

Format

PAUSEERR [remark]

Explanation

You can insert a PAUSEERR subcommand into a batch file to delay the processing of the file. The pause in processing allows you to let an action occur, such as changing disks.

PAUSEERR always sends the following message to the standard error device (the screen), regardless of any file redirection that may be in place:

Strike a key when ready . . .

The optional remark can contain a maximum of 122 characters.

Example

The following example shows a PAUSEERR subcommand in a batch file:

```
COPY a:*. * b:
PAUSEERR Insert new disk in drive A
COPY a:*. * b:
```

When the batch file is run with output redirected to the NUL device, the first diskette in drive A: is copied. Then processing is suspended and this message appears:

```
PAUSEERR Insert new disk in drive A
Strike a key when ready . . .
```

REM

Format

```
REM | ; [remark]
```

Explanation

Use REM to insert remarks in a batch file.

The maximum length of a remark is 123 characters. You can use spaces, tabs and commas as separators. You can also use REM to add space to your batch file for readability.

Note

If ECHO is ON, REM comments appear on the screen as the file is processed. If ECHO is OFF, REM comments are not displayed.

Example

The following is an example of a REM command:

```
REM Archive backup files for security
```

SHIFT

Format

SHIFT

Explanation

SHIFT allows you to change the position of command line parameters (%0 through %9). The parameters on the command line are shifted one position to the left; the parameter in the %1 position moves to the %0 position, the parameter in the %2 position moves to %1, and so on. Each SHIFT command moves all the parameters to the left by one more position.

This lets you write batch files that to access more than 10 replacement variables. Execute the SHIFT command to get past the tenth parameter on the parameter list. It also lets you write batch files that only operate on a single parameter position, for example %1.

Example

Consider the following batch file called **dirlist.bat**:

```
ECHO=off
REM This batch file calls DIR on a number of
REM specified directories or drives
:loop
IF "%1%" == "" GOTO done
DIR /P %1
SHIFT
PAUSE
GOTO loop
:done
ECHO Task complete
```

This example displays directory listings of various specified directories or drives, operating on the %1 position as many times as there are parameters included on the command line. At the command line you could type for example:

```
C>DIRLIST a:\sales b:\sales\foreign\*.doc b:\clients
```

This would produce a listing for each of the specified directories.

10.6 The autoexec.bat file

The **autoexec.bat** file is a special batch file that the operating system runs each time your computer is started. **Autoexec.bat** sets up your computer the way you want it each time the computer is switched on.

If your system supports individual user accounts, there will be a separate **autoexec.bat** file for each user. This will reside in each user's home directory and be run whenever he or she logs in.

For more information on how to set up and modify **autoexec.bat** files, see Chapter 16, "Customizing your system."



|

Commands

THIS chapter contains a complete list and description of the commands available with the operating system.

It also provides an explanation of command syntax and how to edit commands at the command line using the command line editing keys.

Note that, where commands are described in detail in another chapter, only introductory information and the command format is given, with a reference to where to find the full description.

11.1 Command summary

This section contains a table listing all the commands with a brief summary of what they do. The table shows whether commands are internal or external (see Chapter 6, “Working with commands,” for more on internal and external commands).

The /H switch

Remember that each command has a /H switch that displays help text on the command and its parameters. The /H switch should be typed immediately following the selected command, for example `ATTRIB /H`. Instead of /H you can use /. Type /H (or /?) on its own on the command line for a listing of all the available commands.

Table 11-2 Command Set

Command	Explanation	Status
APPEND	Specifies a search path for data files and overlays.	Internal
ASSIGN	Reassigns a drive letter to a different drive.	Internal
ATTRIB	Displays and modifies a file's attributes.	External
AUXPORT	Sets or displays the default serial port.	External
BACKUP	Makes security copies of files, directories and hard disks and diskettes.	External
BANK	Defines if applications will run in expanded or conventional memory.	Internal
BREAK	Controls how you break out of programs running under the operating system, using Ctrl+Break .	Internal
CACHE	Displays the size of cache memory.	External
CHDIR (CD)	Shows the path to, and changes the current directory.	Internal
CHKDSK	Checks the integrity of data on disks and restores corrupted disks.	External
CLS	Clears the display screen.	Internal
COMMAND	Loads a second copy of the default command processor.	External
COMP	Compares files character by character.	External
COPY	Copies or combines files.	Internal

Table 11-2 Command Set (continued)

Command	Explanation	Status
CSPOOL	Controls the print spooler.	External
DATE	Displays and changes the date.	Internal
DEL	Deletes files.	Internal
DELQ	Deletes files but queries you before doing so.	Internal
DIR	Displays the files in a directory.	Internal
DISKCOMP	Compares entire disks of the same format.	External
DISKCOPY	Copies entire disks of the same format.	External
EDITOR	A program used to display write, and change text files.	External
ERAQ	Erases files but queries you before doing so.	Internal
ERASE (ERA)	Erases files.	Internal
EXE2BIN	Converts executable files (.exe) to binary.	External
EXIT	Returns you to a running application.	Internal
FASTOPEN	Increases the speed with which disk files can be accessed.	External
FC	Compares files.	External
FDISK	Formats hard disk partitions or gives information on partitions.	External
FILELINK	A utility used to transfer files between computers via their serial communications ports.	External
FIND	Finds a string of characters within a group of files.	External
FORMAT	Formats diskettes so they can be used.	External
IDLE	Turns dynamic idle detection on or off.	Internal
KEYB	Displays the national keyboard.	External
KEYS	Sets the DOS session switching keys for a terminal while the system is running.	External
LABEL	Creates or modifies a label on a disk.	External

Table 11-2 Command Set (continued)

Command	Explanation	Status
LIMSIZE	Sets or displays the maximum expanded memory available for each application.	Internal
LOCK	Temporarily locks a terminal until the user's password is entered (if security is enabled).	External
LOGOUT	Used to log out if security is enabled.	External
MAIL	Sends and receives electronic mail.	External
MEMSIZE	Sets or displays the maximum conventional memory available for each application.	Internal
MKDIR (MD)	Creates a subdirectory.	Internal
MODE	Sets monitor type, printer type, and COM port.	External
MORE	Causes an application's output to the console to be displayed a screenful at a time.	Internal
NETDRIVE	Displays or sets the status of drives as Local or Remote (network) drives.	Internal
PASSWORD	Changes the login password or manages file passwording.	External
PATH	Sets a search path for commands or batch files.	Internal
PCTERM	Starts the PC Terminal emulator which allows PCs to be used as terminals.	External
PIFED	The Program Information File which defines how an application will be run.	External
PRINT	Prints files.	External
PRINTER	Sets or displays the default printer port.	External
PRINTMAP	Assigns physical printer and auxiliary ports to logical DOS names. Displays default assignments.	External
PROMPT	Modifies the command prompt.	Internal
REBOOT	Command to reboot the system.	External
RECOVER	Recovers corrupted files from disk.	External

Table 11-2 Command Set (continued)

Command	Explanation	Status
RENAME (REN)	Renames files or moves files between subdirectories on the same disk.	Internal
REPLACE	Selectively copies files.	External
RESTORE	Restores files backed up with the BACKUP command.	External
RMDIR (RD)	Removes specified subdirectory.	Internal
SECURITY	Adds users and passwords, if system security has been enabled	External
SET	Inserts strings into the command processor's environment.	Internal
SETUP	A utility used to change system configuration values.	External
SHARE	Dummy utility for DOS compatibility.	External
SHOW	Displays system information such as memory usage and free disk space.	External
SID	A debugging program.	External
SORT	Reads data from the standard input device, sorts and then writes it to the standard output device.	External
STOP	Stops an application on the current station when it cannot be stopped in the usual manner.	Internal
SUBST	Allows you to replace a path with a drive.	Internal
SUSPEND	Allows applications to be suspended when they are running in a DOS session that is not displayed.	Internal
SYS	Transfers operating system files from the source drive to a specified drive.	External
TIME	Displays and changes the time of day.	Internal
TOUCH	Resets time and date stamps on a file or group of files.	External
TREE	Displays a hierarchical path of files, directories and subdirectories.	External
TYPE	Displays the contents of a text file on screen.	Internal

Table 11-2 Command Set (continued)

Command	Explanation	Status
VER	Displays the version number of the operating system in use.	Internal
VERIFY	Switches the VERIFY option on or off. When switched on, VERIFY checks that data are correctly written to disk after each write operation.	Internal
VOL	Displays the disk volume label.	Internal
XATTRIB	Changes file access permissions.	External
XCOPY	Selectively copies groups of files.	External
XDEL	Selectively deletes groups of files and subdirectories.	External
XDIR	Extended directory list with switches to display, for example, files in subdirectories.	External
XSTOP	Stops an application anywhere in the system.	External

11.2 Command syntax explained

The syntax of each command is a combination of the following items:

filename	The command's filename. Some command filenames can be abbreviated, as explained in the text of this chapter. You do not need to include the file extension. To use an external command you must have its corresponding "program file" in your current directory, or in one of the directories pointed to by the path statement.
dirname	Represents a directory name.
filespec	Represents a file specification which can include some or all of the following: [d:] [dirpath] filename [.ext] [;password] where: <i>d:</i> is a drive, for example C:. <i>dirpath</i> is the location of a file in the directory structure, for example \work\ or \. <i>filename</i> is the 1-8 character name of a file. <i>.ext</i> is a one- to three-character file extension, for example .bat. <i>,password</i> is a one- to eight-character password.
wildspec	This is the same as a filespec, but it indicates that you can use the wildcard characters ? and * in filenames and extensions to specify groups of files.
@	If you see the symbol @ in front of a filespec, it means you can insert a filename containing a list of files which you want the command to operate on. You can create a text file (using EDITOR) containing several different filenames, and you can even use wildcard characters in the names. Note that each filename in your text file must be placed on a new line. Note also that you

	cannot use wildcard characters in the filelist name.
device	Represents a peripheral device, for example <code>prn:</code> .
[]	Square brackets mean an optional clause in a command. Type what is shown in the brackets, do not enter the brackets into the command line. For example for <code>[/C]</code> , you type <code>/C</code>
	A vertical bar means either/or, for example <code>ON OFF</code> . Choose one or the other of the separated items. Do not type the vertical bar.
...	Three periods means you can repeat an item, for example if a filespec or wildspec is followed by <code>...</code> , it means you can have multiple file or wildspecs on one command line. So instead of typing two commands like this: <code>FIND error *.c</code> and <code>FIND error *.h</code> you can put the wildspecs <code>*.c</code> and <code>*.h</code> on the same command line, as follows: <code>FIND error *.c *.h</code>
n	This means enter a number within a specified range. The command explanation gives the range of valid numbers.

Note

All punctuation (excluding the special cases listed above) should be entered as shown.

11.3 Editing commands at the command line

The operating system has powerful command line editing features to save you effort when typing commands. In particular, it can store in memory the commands you have issued recently. This saves you for example having to retype a command line if you make a typing error. By using certain keys you can redisplay the command, edit it and reissue it.

Say you mistype the ERASE command:

```
C>efase myfile.doc
```

The error message:

```
Command or filename not recognized
```

is displayed. Although you have entered the command by pressing **Enter**, it is not lost, but stored in a special area of memory called a *buffer*. You can use the command line editing keys to correct the command and then re-enter it:

Press **↑** To display the command line you just typed.

Press **←** To move the cursor using the left arrow key until it is on top of the *f*.

Press **Del** To delete *f* from the stored line.

Type *r* To enter *r* into the stored line.

Press **↵** To re-enter the correct command.

Similarly the command:

```
C>erase myfile.doc
```

can be corrected by the following keystrokes:

Press **↑** To display the previous command lines.

Press **←** To move the cursor until it is on the *l*.

Type *i* To add *i* to the display.

Press **↵** To re-enter the correct command.

Command line editing keys

Table 11-2 lists the command line editing keys and their meanings.

Table 11-2 Command Line Editing Keys

Keystrokes	Meaning
← or Ctrl+S	Move one character to the left.
→ or Ctrl+D	Move one character to the right.
↑, F3 or Ctrl+E	Move to the previous command in memory.
↓ or Ctrl+X	Move to the next command in memory.
Enter or Ctrl+M	Enter the command line
Backspace or Ctrl+H	Delete the character to the left of the cursor.
Del or Ctrl+G	Delete the character at the cursor.
Ins or Ctrl+V	Switch insert/overstrike mode on or off.
Ctrl+A	Move one word left.
Ctrl+F	Move one word right.
Home or Ctrl+Q	Move to the beginning of the line.
Ctrl+W	Move to the end of the line.
Ctrl+T	Delete a word.
Ctrl+Y	Delete a line.
Ctrl+B	Delete to the beginning of the line.
Ctrl+K	Delete to the end of the line.
Ctrl+R	Start command line search (see *).
Ctrl+_	Turns command line search mode on or off. The default is off. (See *.)

*** Searching for commands**

Ctrl+R displays the last matching command. You can then enter the displayed command again, or edit it before pressing **Enter**.

Ctrl+_ switches on a more sophisticated version of command line search. When it is active, each time a character is typed, the last command that matches that character is displayed. For example, if you type **D** then the last command you entered that started with the letter **D** is displayed. If you type **C**, then the last command you entered which began with the letter **C** is displayed.

Command line search remains on until you type **Ctrl+_** again.

Control key combinations

There are certain control key combinations which can be used to carry out particular tasks, they are listed below:

Key combination	Function
Ctrl+C or Ctrl+Break	Stops commands when they are running. The processing of the command is stopped, and the command prompt returned.
Ctrl+P	Switches between echoing text to the screen, or to a printer.
Ctrl+S	Suspends scrolling. To start scrolling again, press Ctrl+Q .

11.4 Command reference

There follows a reference section containing descriptions of each of the operating system commands.

APPEND

Format

```
APPEND [/H] [d:]dirpath [;[d:]dirpath...] [/X]
```

Explanation

APPEND enables programs to look in other specified directories for files that it cannot find in your current directory. It is similar to the PATH command. PATH is used to search for .com, .cmd, .exe and .bat files. APPEND is used when programs want to search for any other type of file.

The APPEND command is generally used with older-style DOS applications which do not support pathnames. Normally you should not need this command, unless your applications documentation advises you to use it.

Command switch

/X	Causes APPEND to work with search and exec functions.
----	---

Before you enter an APPEND command to specify a search path, you can issue APPEND with the /X switch. This causes APPEND to work with programs that use the Search First, Find First, and Execute functions. Alternatively, you can add /X to your command line in the ordinary way.

Caution	Use APPEND with care; it can lead to problems when used with certain application programs. Some applications cannot use the APPEND command; you will find that they read files successfully from directories other than the current directory, but they can only create files in the current directory. This results in two copies of files: an edited version on the current directory and the unedited version left unchanged in the original directory.
----------------	--

APPEND with pathnames

APPEND followed by pathnames is used to specify a search path. Once entered, APPEND specifies where the operating system is to look for files if it cannot find them in the current directory. You can specify several search directories provided

each is separated by a semi-colon (;). The total length of the APPEND command cannot exceed 128 characters.

To return to the default value of no search path, type:

```
C>APPEND ;
```

APPEND and BACKUP

Before you use the BACKUP command you should cancel the APPEND search string by entering:

```
C> APPEND ;
```

Examples

The following command sets up a search path such that if a program cannot find the file you want in your current directory, it looks in the directory **c:\spreads\cashflow**.

```
A>APPEND c:\spreads\cashflow
```

The following example extends the search path so that the program also looks in the directory **\oldones** on the A: drive.

```
A>APPEND c:\spreads\cashflow;a:\oldones
```

ASSIGN

Format

```
ASSIGN [/H] [drive_letter=drive_letter] [/A]
```

Explanation

ASSIGN causes commands that would access a certain drive to access a different drive. It is mainly used when you have a program set up to access a certain drive like drive A: but the information to access is stored on a different drive (for example drive C:). You can use ASSIGN rather than re-installing or modifying your program.

To cancel an ASSIGN statement, simply type ASSIGN

Command switches

/A	Displays the current status and assignment.
----	---

Remarks

Do not use the following commands when an ASSIGN command is in effect:

BACKUP
DISKCOMP
DISKCOPY
FORMAT
LABEL
RESTORE
SUBST

Examples

The following ASSIGN command reassigns drive A: as drive C::

```
C>ASSIGN a=c
```

So for example, if you type:

```
C>DIR a:
```

you will now see a DIR listing of the C: drive. Typing ASSIGN /A will give you the current assignment made, in this case:

```
A: =>C:
```

If you type :

```
C>DIR c:
```

you will also see a DIR listing of the C: drive.

Caution	The above example means that any command issued that includes the drive letter A is read as if it was the letter C. This means drive A: is inaccessible to you. In general, do not use ASSIGN unless you are running a specific program that requires it.
----------------	---

The following command cancels any ASSIGN statements made:

```
C>ASSIGN
```

ATTRIB

Format

```
ATTRIB [/H] [+|-A] [+|-H] [+|-R] [+|-S] [@]wildspec...  
[/P] [/S] [/U:name]
```

Explanation

ATTRIB displays a file's attributes, and can be used to switch these attributes on or off. The attributes included are the archive, hidden, read-only and system attributes.

+A attribute

The archive attribute (+A) shows that a file has been modified. The ATTRIB command enables you manually to switch the archive bit on or off for any given file. This is useful when you use the BACKUP switch /M which copies all files with the archive bit set.

+H attribute

The hidden attribute (+H), when switched on, hides files from view in standard directory listings.

Note

Both COPY and XCOPY have switches that allow you to copy system files.

+R attribute

A file that is set to be read-only (+R) cannot be erased, renamed, or altered in any of the normal ways. Setting the read-only attribute on a file should protect it from being inadvertently altered. However, the REPLACE, TOUCH and XCOPY commands all have /R switches that allow you to alter read-only files.

+S attribute

The system attribute (+S), when switched on, prevents DIR from displaying the file and COPY from copying it unless the /S switch is specified. It is used for the system files that you do not need to see in everyday work.

Note

Where system security is enabled, use the XATTRIB command for enhanced file access support.

Command switches

+A	Sets the archive attribute (-A switches it off).
+R	Sets the read-only attribute (-R switches it off).
+H	Sets the hidden system file attribute. (-H switches it off).
+S	Sets the system attribute (-S switches it off).
/P	Pauses after each screenful in a long display.
/S	Sets the file attributes in the subdirectories associated with the chosen directory.
/U: <i>name</i>	Limits the effect of the ATTRIB commands to the files belonging to the user <i>name</i> or to the group <i>name</i> .

Examples

The following example switches on the archive attribute for all the files with the extension **.new** in the current directory. You can then use a command like XCOPY /M to copy all the files marked for archiving:

```
A>ATTRIB +A *.new
```

The following command switches off the read-only attribute for all the files with the **.bas** extension in the current directory, and any associated subdirectories:

```
A>ATTRIB -R *.bas /S
```

The following command switches on the read-only attribute for all the files with the **.bat** extension in the current directory which belong to the user ISMITH:

```
A>ATTRIB +R *.bat /U:ISMITH
```

AUXPORT

Format

AUXPORT [/H] [n]

Explanation

The AUXPORT command selects one of the auxiliary ports. The number of auxiliary ports in your system depends on the system configuration. Each DOS session can have a different AUXPORT assignment.

When you enter AUXPORT with no parameters, it displays your current DOS session's AUXPORT setting. If you enter AUXPORT with a parameter, it changes the status of your current DOS session and displays the new AUXPORT assignment.

Examples

The following example displays your DOS session's current AUX number:

```
A>AUXPORT
COM1      AUX 0
```

The following example sets the AUX number of your current DOS session to 1:

```
A>AUXPORT 1
COM1      AUX 1
```

BACKUP

Format

```
BACKUP [/H] source_drive [filespec] destination_drive
[/A] [/D:date] [/F] [/L[:filespec]] [/M] [/S]
[/T:time] [/U:name]
```

Explanation

BACKUP makes backup copies of your files.

You can backup whole hard disks and diskettes, or directories, groups of files, or even single files. The backed-up files must be restored using the RESTORE command before they can be used by the system.

BACKUP and RESTORE programs are compatible with DOS BACKUP and RESTORE programs, version 3.30 or later. Files backed up with BACKUP can be restored with a different DOS RESTORE and vice versa.

Caution Password-protected files and subdirectories are NOT backed up, unless the global password is set before running BACKUP, and the files have been protected with the same password.

With system security

If system security has been enabled, the /U option allows you to specify:

- ☐ that you want to record the owner of each file that is backed up. (It will then be restored to the correct user, when you use the RESTORE command.)
- ☐ which user's or which group's files to back up.

Note

Unless you are logged in as the Superuser, you can only back up files to which you have read access.

Command switches

/A	Adds new files to the backup disk without erasing existing files (the default is to overwrite all the existing information on the backup disk).
/D:date	Backs up only the files that have been modified since the specified date. The date format you use is governed by the COUNTRY command in your cconfig.sys file, but is usually the one commonly used in your country: month/day/year in the US, day/month/year in most European countries, and year/month/day in Japan.
/F	Formats the destination diskette before copying the files.
/L[:filespec]	Creates a file recording the date and time of backup and number of the backup diskette on which each file is stored. The default filename for this backup log is backup.log, though you can specify another filename. The log is placed in the root directory of the source disk. If you specify a name for the log file that already exists on the root directory, the new log information is appended to the end of the existing log file.
/M	Backs up all the files on the disk modified since the last backup. (This switch actually copies all the files for which the archive bit is set. You can set the archive bit manually by using the ATTRIB command.)
/S	Backs up the current directory and all the subdirectories associated with it. You must be in the root directory and specify /S if you want to backup an entire disk. The default is to backup only the <i>current</i> directory.

Command switches (continued)

<i>/T:time</i>	Backs up only the files that have been modified since the specified time. The format to use is hh:mm:ss.
<i>/U:name</i>	If you enter /U without a <i>name</i> the owner of each file is recorded as the backup is taken. Enter a <i>name</i> if you only want to back up files belonging to one owner or group.

BACKUP reports any errors using the codes shown in Table 11-3.

Table 11-3 BACKUP Error Codes

Code	Explanation
0	Normal completion.
1	No files found to backup.
2	Some files not backed up due to file sharing conflict.
3	Backup ended by Ctrl+Break command.
4	Backup ended by an error.

This feature can be useful in creating batch files with an `IF ERRORLEVEL` statement in them to improve the process (see Chapter 10, "Batch processing").

Remarks

Do not use the BACKUP program with disks affected by an `ASSIGN` or `SUBST` command.

When you make a backup, label and number all your diskettes consecutively to make it easier when you come to restore them using the `RESTORE` command.

If you do not want files overwritten on your backup diskettes, specify the `/A` switch in the command line.

Examples

The following command backs up the entire hard disk onto diskettes. When each diskette is full, you are prompted to remove it and insert the next one:

```
C> BACKUP c:\ a: /S
```

Note

To backup a full 20 Mbyte hard disk you may need up to 35 diskettes, though this number varies with the capacity of your diskette drive.

The following command backs up all the **.doc** files that have been modified since the last BACKUP on directory \ww and on the subdirectories associated with it:

```
C> BACKUP c:\ww\*.doc a: /S /M
```

The following command backs up all the **.doc** files belonging to user FRED that have been modified since the last BACKUP on directory \ww and on the subdirectories associated with it:

```
C> BACKUP c:\ww\*.doc a: /S /M /U:fred
```

BANK

Format

BANK [/H] [=]ON|OFF]

Explanation

The BANK command controls how a program runs in banked (paged) memory. The default setting is BANK ON.

If you enter BANK OFF, the operating system cannot page the program in and out of conventional memory. This means that the memory used by the program is removed from the total memory available for paging. A BANK OFF setting can thus limit system performance severely.

If you enter BANK on its own the current setting is displayed.

Remarks

If you enter BANK as a command it only affects applications *in the current DOS session*.

The PIFED information for a particular application overrides the BANK command setting.



BREAK

Format

BREAK [/H] [[=]ON|OFF]

Explanation

When BREAK is switched ON it allows you to stop a running program by holding down the **Ctrl** key and pressing **Break** or **C**. The program will then stop the next time it tries to call any DOS function, such as accessing the disk.

If BREAK is OFF (the default value), a **Ctrl+Break** or **Ctrl+C** command only takes effect when the program you want to stop is performing standard input or output operations such as printing or outputting to the console.

If you type BREAK on its own, you see the current BREAK value.

Note

Entering BREAK at the command line overrides the BREAK setting in the **cconfig.sys** file *for the current DOS session*.

Examples

The following command switches BREAK to ON. BREAK remains ON *in the current DOS session* until you issue another BREAK command or reboot your system.

A>BREAK ON

The following command shows the current status of the BREAK command.

A>BREAK

CACHE

Format

CACHE [/H]

Explanation

Accessing data on a hard disk can be time consuming. To speed this process up you can set up an area of RAM memory which acts as a buffer (the *cache*), storing a copy of data frequently read from disk. Every time your applications attempt to access information on disk, a check is first made to see if a copy of the data is already in the *cache*. If it is, the copy in the cache is used instead. This can be much faster than accessing the disk itself. Caches can be very useful if your work involves applications that need to access the same data over and over again, like a database or spreadsheet.

Cache is set up using the INSTALL or SETUP command. At the command line, you can display the current status by typing:

A>CACHE

If cache has been installed, you see a display below the copyright and title lines, similar to this:

```
1 drive cached in 123 track buffers
2091 K Bytes of Extended Memory used
69% of CACHE in use with a 74% hit rate.
```

The report on CACHE shows the number of drives and tracks cached, the amount and type of memory used, and the efficiency of the cache.

Remarks

CACHE is always set up in extended memory.

CHDIR (CD)

Format

CHDIR [/H] [d:][[[d:]dirpath]

Explanation

The CHDIR (change directory) command allows you to change the current directory on a drive and display the current directory path. The short form of the CHDIR command is CD.

Where no parameter is specified, CHDIR displays the directory path leading to the current directory. Where only a drive is specified, the current directory of that drive is displayed.

If a directory path is specified, CHDIR makes the last directory in the path the current directory, for the current drive or the drive specified.

If the directory path specified is on the current drive, you are immediately moved to the new directory. If, on the other hand, the directory path relates to a drive other than the one you are in, you remain where you are. However, if you now move to the drive you have just specified, you will find yourself in the new current directory.

If the directory you are currently in is not part of the path to the directory you want, you must specify the complete path from the root, preceding the first directory with a backslash.

To help you move between directories, CHDIR responds to the special characters \ (backslash) and .. (two periods).

Enter a backslash alone with CHDIR and you are returned to the current or designated drive's root directory. For example, the command:

A>CHDIR c:\

makes the root directory on drive C: the current directory.

Enter two periods (..) with CHDIR and you are returned to the parent directory, that is, the next directory up in the direction of the root.

For example, if the directory **bonus** in the path **accounts\wages\bonus** is the current directory, the command:

A>CHDIR ..

Special characters

makes **wages** the current directory.

Examples

The following command makes **personal** the current directory. The backslash that precedes **clients** indicates that the path to **personal** begins from the root of drive C:, not from the current directory:

```
C>CHDIR \clients\personal
```

The following command makes the drive C: root directory the current directory:

```
C>CHDIR \
```

The following command makes **memos** the current directory on drive A:. All references to A: now access this directory:

```
C>CHDIR a:\wordproc\memos
```



CHKDSK

Format

```
CHKDSK [/H] [d:wildspec] [/B] [/C] [/D] [/F] [/L]  
[/M] [/P] [/R] [/S] [/V]
```

Explanation

The CHKDSK (check disk) command scans a disk and checks it for two things: space allocation and spacing errors. It then produces a disk status report that shows how much total space is left on the disk, the amount of space in use in bytes and the number of files (including hidden files). It also reports on any errors found. If you do not specify a disk drive, the current drive is assumed.

A typical disk status report looks something like this:

```
Volume MYDISK created 2 Apr 1989 13:44
```

```
362,496 bytes total disk space  
246,784 bytes in 20 user files  
115,712 bytes available on disk
```

```
543,184 bytes available
```

It is a good idea to check your disk(s) on a regular basis to keep account of how much space you have used, and to check for any errors.

Use output redirection to a file, if you wish to have a permanent record of the report.

Error correction

CHKDSK will not correct any errors found unless you ask it to by specifying the /F switch. This is because correcting errors often involves altering the File Allocation Table and so may mean loss of data. (The File Allocation Table, or FAT, is a kind of index used to locate your files and allocate space to create new files.)

As the CHKDSK program analyzes the disk it looks for extracts of files that have got “lost” and “mixed up” so the system can no longer tell which file they came from. These are called *lost clusters*. If any lost clusters are found CHKDSK displays a message like this:

```
Five lost clusters found in 2 chains.  
Convert lost chains to files (Y/N)?
```

Answer Y for Yes and the lost clusters are collected into files in the root directory with the name **filennnn.chk** (nnnn = numbers starting from 0000 to a maximum of 9999).

You may be able to examine these files in EDITOR or using the TYPE command (unless they are in binary format), to see if they contain any valuable data. Then recover any data you need, and delete the file(s).

Answer N for No and the clusters are freed for use when new files are created.

Command switches

/B	Mark bad clusters. The /B switch searches all the files on a disk to locate any bad clusters. This switch can be used to check the integrity of the file system, (for example following errors from other CHKDSK switches).
/C	Displays cluster numbers of all the bad files.
/D	Locates erased directories on a disk so that in the event of disk corruption, you can use a suitable disk editor to restore individual files and directories.
/F	The /F(ix) switch allows CHKDSK to write corrections back to the disk. If you do not specify /F, CHKDSK does not modify the disk when errors are found. The default is read-only (that is, do not fix).
/L	Rebuild cluster links. /L attempts to rebuild a corrupted File Allocation Table (FAT) if for some reason it has become corrupted. You must zero any corrupted areas on the disk before using this switch.
/M	Verifies the disk for bad clusters and maps bad clusters. This switch is used to mark bad clusters after formatting. It can be used if a file has become corrupt, or if errors were encountered writing to a specific location on a disk suggesting a bad cluster. Having marked a cluster as bad, the system does not attempt to allocate that cluster again.

Command switches (continued)

/P	Displays parent clusters for all directories. This switch is used to examine a directory tree, when checking the integrity of a disk.
/R	Recover root directory. /R recovers lost directories that were contained in the root directory. You must remove any corrupted files in the root directory.
/S	Shows the actual file space. This is a total of actual bytes taken up by the files on the disk, as opposed to clusters taken up on the disk, that is, the actual number of bytes of file space occupied rather than disk space.
/V	Verbose display. /V displays additional information while CHKDSK is running and may be useful when trying to locate corrupted data on a disk.

Remarks

CHKDSK does not work on drives used in ASSIGN or SUBST commands.

CHKDSK does not work on any drives that have open files on them.

CHKDSK also does not work on networked drives. If you need to run CHKDSK on a remote drive, when no other user is on the system, use the NETDRIVE command to designate the drive as a local drive. Then use CHKDSK.

CHKDSK followed by a file specification will check that the specified files are allocated contiguous clusters on disk. Any file that consists of discontinuous clusters is displayed.

You can check diskettes as well as hard disks using CHKDSK; however, make sure the diskette is in the diskette drive before you issue the command.

Examples

The following example checks the diskette in drive A:, and writes any corrections back to the diskette:

```
C>CHKDSK a: /F
```

The following example sends your disk status report directly to a file called **chkdsk.txt**:

```
C>CHKDSK >chkdsk.txt
```

(Do not use the /F switch if you redirect CHKDSK output.)

CLS

Format

CLS [/H]

Explanation

CLS clears the display screen. If foreground and background colors were selected previously, the colors will be unchanged. Otherwise, the screen is set to white characters on a black background. It is useful if you want to start with a fresh screen.

Example

To start with a fresh screen, from the command prompt type:

```
C>CLS
```

COMMAND

Format

COMMAND [/H] [d:][dirpath] [/Cstring] [/E:nnnnn] [/P]

Explanation

COMMAND starts a copy of the operating system command processor **command.com**. **Command.com** processes the commands you issue. In the case of internal commands, **command.com** can execute the commands directly. In the case of external commands and other applications, it arranges for the appropriate programs to be loaded from their files on disk. The COMMAND options are most commonly used by applications when they execute DOS internal commands.

The command processor executes in an *environment* which is the collection of all system variables such as PATH and PROMPT. When a command processor is started, it inherits the environment from its *parent*. Any changes made to the new environment (for example using SET or PROMPT), do not affect the primary environment.

The command processor resides in memory in two parts: the resident part which remains in memory all the time, as long as the operating system is running, and the transient part which may be overwritten by some external commands or applications. If this happens, then the resident part can reload the transient part from the disk file **command.com** in the directory specified by the *d:* and *dirpath* options, when the application completes, and before the next command is processed.

The loaded copy of the command processor remains in memory until you issue an EXIT command.

You can specify a device name in the command syntax if you want to use a different device for input and output, e.g. AUX.

If you want to run terminate-stay-resident programs, you should run **command.com** first.

Note

COMMAND entered on its own causes a new copy of the command processor to be loaded.

Command switches

<i>/Cstring</i>	Makes the command processor carry out the command(s) or program(s) specified in the string and then exit and return to the primary command processor. If this switch is used with other switches it must be the last one specified on the command line.
<i>/E:nnnnn</i>	The size of the environment in bytes where <i>nnnnn</i> is in the range 512 (minimum) to 32,751 (maximum). If you specify a value outside this range, the command defaults to the minimum value.
<i>/P</i>	Results in this copy of the command processor being fixed permanently in memory. The EXIT command is disabled.

Example

To set the environment size when you load the command processor enter:

C>COMMAND /E:1024

COMP

Format

COMP [/H] [wildspec wildspec] [/A] [/M:n]

Explanation

This command compares two files character by character and reports any character mismatches. COMP also reports if the End-Of-File (EOF) character is missing from any file that it examines.

Wildcards are allowed in the file specification. If you leave out the two filenames altogether, COMP assumes a *wildspec* of *.* and prompts you for the source and destination.

Command switches

/A	Displays the reported mismatches in ASCII format. (The default is to display them in hexadecimal format.)
/M:n	Specifies the maximum number of mismatches before COMP terminates automatically; 10 is the default, 0 allows an unlimited number of mismatches.

Examples

The following command compares all the **.doc** files in directory **c:\address** with files of the same name on the root directory of a diskette in drive A:

```
C>COMP \address\*.doc a:\
```

The following command compares the file **today.doc** with the file **lastweek.doc** to see if the two contain identical information. The two files are both in the same directory, which is the current directory. Only five mismatches are allowed before COMP terminates automatically.

```
C>COMP today.doc lastweek.doc /M:5
```

COPY

Format

```
COPY [/H] device|wildspec [+wildspec]...[/A]/[B]/[C]  
[/S]/[V]/[Z] device|filespec [/A]/[B]/[C]/[S]/[V]/[Z]
```

Explanation

The main function of the COPY command is to copy one or more files to another location either on the same disk or another disk. It can also be used to combine files together into one file, to create files and to send files to a printer.

COPY can be used with password-protected files but note:

- ☐ COPY does not prompt for a password.
- ☐ The destination copy of the file will not have password-protection unless you assign a password with the PASSWORD command.

If system security has been enabled, you will only be able to copy files to which you have at least read access.

COPY has several switches you can add to both source and destination specifications:

Command switches

/A	Causes the file to be treated as an ASCII (text) file. When you specify the /A switch with the source specification, the file contents are copied up to, but not including, the first Ctrl+Z (End-Of-File) character. Remaining file characters are not copied. Using /A with a destination specification adds a Ctrl+Z as the last character in the file, which is required by some word processors.
/B	When applied to the source <i>wildspec</i>, /B causes the file to be treated as a binary file; the entire file is copied regardless of any Ctrl+Z character found in the file.

Command switches (continued)

When you use /B with your destination *filespec*, no end-of-file character (**Ctrl+Z**) is added.

Note

The /A and /B switches only need to be entered when you are combining ASCII and binary files. These parameters are not necessary to copy data, program or batch files.

/C	Asks for confirmation before copying each file. If you are combining multiple groups of files, it confirms each group by first source name.
/S	Includes system and hidden files in the copy.
/V	Causes COPY to compare the destination file with the source file contents to verify the accuracy of the copy. Because of the comparison, it takes longer to complete the copy with the /V switch.
/Z	Zeros the top bit of every byte in the destination file. This switch should only be used with ASCII files. (It is designed for use with older-style word processors which use the eighth bit in every byte to mark beginnings and ends of words. This means text written in such a word processor would be unintelligible in a newer-style word processor. The /Z switch combats this problem.)

Copying files

Wildspec is the source (the file to be copied) and must always be given. *Filespec* is the destination. The source and destination file specifications are comprised of the complete path, filename and filename extension.

If you omit the destination *filespec*, the “copy” is created in the current directory of the disk in the default drive, with the same name as the source file. You cannot copy a file to itself.

If you use wildcards in the source *wildspec*, the names of the files are displayed as they are copied.

If you type:

```
C>COPY *.* reports
```

all the files in the current directory would be copied to the directory **reports**. However, if the directory did not exist, ALL the files would be copied to a single file called **reports** in the current directory.

Specifying the
destination filespec

If *filespec* is a drive name only, for example:

```
C>COPY file.txt a:
```

the file is copied to the specified drive with its original name.

WARNING If you enter a destination file specification that matches one already on the destination drive, the copy *overwrites* the existing file.

If *filespec* is a filename only, for example:

```
C>COPY file.txt new.txt
```

the original file is copied to a file on the current drive with the name given. Note that this is a method you can use COPY to rename a file or files.

If *filespec* includes a drive name:

```
C>COPY file.txt a:new.txt
```

the original file is copied to the file on the designated drive.

Note

To copy entire directories and subdirectories, use the command XCOPY.

Examples

To copy the file **myfile** from the current drive to drive B:

```
A>COPY myfile b:
```

To copy the file **myfile.typ** on drive B: to the current drive A:

```
A>COPY b:myfile.typ
```

The next command duplicates the file **myfile.typ** in the same directory and names the copy **yourfile.typ**:

```
A>COPY myfile.typ yourfile.typ
```

You can copy a file from one drive to a different drive by specifying drive designations *before* the filenames. This example copies **phone.lst** from drive B: to drive C:, and verifies the destination file against the source file for an exact match:

```
A>COPY b:phone.lst c:phone.lst/v
```

Combining files

To combine (or "concatenate") a number of source files into a single destination file, type COPY followed by the filenames to be combined, with a plus sign between them. Enter a space between the two combined files and the destination file:

```
COPY file1.txt+file2.txt result.txt
```

The files are combined in the order in which they are listed, or if there is a wildcard, in the order they are encountered in the directory. The destination file is always the last file named.

Caution	If the destination file exists already, it is overwritten.
----------------	--

You should not combine files if one of the source files has the same name and extension as the destination file.

Example

To combine the file **myfile.typ** on drive A: with **yourfile.typ** on drive B:, copy the file to drive C:, and name it **result.typ**:

```
A>COPY myfile.typ+b:yourfile.typ result.typ
```

Copying files to a console or printer

You can use COPY to display or print a file. To do this, use the form:

COPY wildspec device

COPY can be used instead of TYPE to display or print a file. To perform this kind of copy, enter CON or PRN as the destination. This command is valid only for text (ASCII) files, not application or program files.

The printer and console device names are shown in the table that follows.

Table 11-4 COPY Device Names and Descriptions

Name	Description
CON	Indicates the screen when it is the destination and the keyboard when it is the source device.
LPT n	Indicates the destination printer where n is 1, 2 or 3.
PRN	Indicates the current printer.
NUL	Indicates a dummy output device (usually used for program testing).
COM n	Indicates the serial ports where n is 1, 2, 3 or 4.
AUX	Indicates the current auxiliary port.

Examples

To read the file **myfile** from drive D: and display it on the console:

```
A> COPY d:myfile con
```

To print all files with the file extension **.txt** on the current drive:

```
A> COPY *.txt prn
```

**Creating a file with
COPY**

You can create a text file using COPY by specifying the keyboard as the source device:

C> COPY CON filespec

When you specify CON as the source device, every character you type in is recorded in the destination file you specified, except line-editing characters such as **Ctrl+H**.

Press **Ctrl+Z** when you have finished typing your file, and close it by pressing **Enter**.

Examples

To create a file called **showme** with the text “this is my file”, type:

```
A> COPY CON showme
    this is my file
Ctrl+Z
```

After you have pressed **Ctrl+Z** to add the End-Of-File marker (which says this is where the file finishes) and then pressed **Enter**, the command prompt is returned to the screen.

The file **showme** can now be displayed by typing:

```
A> COPY showme con
```

I

CSPOOL

Format

```
CSPOOL [/H] [START|QUIT|STATUS [c]|PRIORITY n p]  
STOP [n] |
```

Explanation

The set of CSPOOL commands that control the print spooler, CSPOOL, are listed in the table that follows. For more information, see Chapter 13, “Printing.”

Table 11-5 CSPOOL Commands

Command	Description
START	Starts the spooler. If you decided to use spooling when you installed the system, or later using SETUP, the spooler will be started automatically each time you restart the system. You will only need to type this command if you stopped the spooler during a session using CSPOOL QUIT.
QUIT	Removes the spooler from the system. This command should be executed before the system is turned off. Any files in the queue are printed the next time the spooler is started. Equally, if a file was interrupted, it is reprinted from the beginning.
STATUS	Displays a list of jobs with their status. The jobs in the queue, who queued them and their priority are listed. If you use the option C, the display of the spooler status is continuously updated until you press Enter .
PRIORITY	Allows the priority of a queued job to be changed. Jobs are printed in order of priority; those with the same priority are printed in the order in which they were submitted. <i>n</i> is the job number. <i>p</i> is the priority, 1 being the highest and 10 the lowest.

Table 11-5 CSPOOL Commands (continued)

Command	Description
STOP	Deletes a print job; either one in the queue or the job that is currently being printed. CSPOOL STOP on its own lists jobs, and asks which one you want to stop.

CSPOOL on its own displays help information about the different CSPOOL commands.

Examples

The following command deletes job number six from the spooler queue.

CSPOOL STOP 6

To change the priority of job number six to the highest priority so that it is printed before jobs other jobs type:

CSPOOL PRIORITY 6 1

DATE

Format

DATE [/H] [nn-nn-nn]

Explanation

The **DATE** command displays or sets the month, day, and year.

Your computer has its own battery-backed clock, so it is not normally necessary to set **DATE** or **TIME** each time you start the computer or reboot the operating system.

The internal clock provides:

- ☐ Continuous access for you and your applications to the date and time.
- ☐ Time and date stamping facilities. (Time and date stamping records the current time and date whenever you create or update a file.)

Use **DATE** to display the current date. When you enter:

A> **DATE**

the current day and date are shown and you are prompted to enter a new value.

Press **Enter** to keep the current values or enter a new date. A new value effects all DOS sessions until the system is rebooted.

To set the date from the command line, type **DATE nn-nn-nn**, where the date is expressed in the format employed in your country.

The *nn-nn-nn* parameter must be made up of values *mm*, *dd* and *yy*, in one of the following orders:

mm-dd-yy

dd-mm-yy

yy-mm-dd

where:

- ☐ *mm* is a one- or two-digit number from 1 to 12 indicating the month.
 - ☐ *dd* is a one- or two-digit number from 1 to 31 indicating the day.
 - ☐ *yy* is a two- or four-digit number indicating the year.
-

You can use either a hyphen (-) or a slash (/) to separate your entries.

If you enter a date that is incorrect or out of range, you see:

```
Invalid date specified
Enter date:
```

Enter the date as described or press the **Enter** key to keep the previous value.

Remarks

The way the date is displayed, and the way any new date entry is interpreted, should match the method used in the country in which your system has been configured. The US form is *mm-dd-yy*, the European is *dd-mm-yy* and the Japanese, *yy-mm-dd*. You can change your configuration using the **SETUP** program or by making a **COUNTRY** entry in the **cconfig.sys** file (see Chapter 16, “Customizing your system”).

DEL

Format

DEL [/H] wildspec [/C] [/S]

Explanation

DEL deletes the specified file or files. The file specification can identify a particular file, or it can indicate a group of files through the wildcard characters * and ?.

After you erase a file, you cannot retrieve it, so be very careful when you use wildcards. Generally, DEL does *not* query you before erasing files. However, it does query you when you use the asterisk (*) wildcard for both the filename and the extension (*.*). For *.* it displays the following message:

Are you sure (Y/N)?

Type Y to erase all files in the directory; type N to cancel the command.

Command switches

/C	Prompts you to confirm deletion.
/S	Includes system files.

Example

The following command erases all the **.bak** files on the current directory without querying you:

A>DEL *.bak

DELQ

Format

DELQ [/H] wildspec [/S]

Explanation

The DELQ command deletes the specified file or files. It queries you to make sure you want to erase each file.

DELQ is most useful when used with the wildcard characters * and ? to erase an entire class of files. Instead of erasing every file that matches the file specification, DELQ lists the specified files individually, followed by a ? prompt, to which you respond either with a Y for Yes (erase the file) or with an N for No (do not erase the file).

Command switches

/S	Includes system files.
----	------------------------

Example

The following command gives you the choice to delete each of the .txt files in the current directory:

A>DELQ *.txt

DIR

Format

DIR [/H] [wildspec] [/A] [/C|/R] [/D] [/L] [/N] [/P] [/S] [/W]

Explanation

DIR displays the contents of a disk directory. DIR with no parameters displays all files other than system and hidden files, and all subdirectories in the current drive's current directory. Files with the system attribute are not shown. (See the ATTRIB command for a description of attributes.) Subdirectories are separated from each other in the command line by a backslash (\).

You can search for a particular file, for example:

A>DIR myfile.gtm

DIR will search the current directory for **myfile.gtm**. If it cannot find the file, this message is displayed:

File not found

DIR shows the contents of the current drive's or the designated drive's current directory unless you specify another drive or path.

You can select a file or class of files to display by entering a specific or a general file specification using the wildcard characters * and ?.

The DIR display shows the following information about the current directory:

- ☐ The drive selected.
 - ☐ The volume label of the current drive.
 - ☐ The full path used for file search.
 - ☐ The list of file names.
 - ☐ The number of files found.
 - ☐ The amount of storage space left, in bytes.
 - ☐ Whether any system files were omitted from the display.
-

The DIR switches can be entered individually or in combination, and in any order.

Command switches

/A	Displays all files.
/C or /R	Used with other switches to make them the default for the next DIR command issued, for example DIR /C /W.
/D	Displays files where the system attribute is reset. This is a default switch.
/L	Includes size, date, and time; the default.
/N	Allows you to return to the default paging switch.
/P	Pauses at the end of each full page. The default is no paging.
/S	Displays files with the system attribute set.
/W	Displays the names of files in a wide format, with backslashes (\) before directories.

Examples

The following command displays all files and subdirectories in drive C:'s current directory with the system attribute:

```
A>DIR c: /S
```

The next command displays all files and subdirectories with the **.bat** file extension in drive A:'s current directory. It uses the "Long" format, giving information on size, and date and time stamping. The display pauses with each full screen. Files with the system attribute set are included and the switches remembered for the next DIR command without switches. (The filespec is not remembered.)

```
A>DIR *.bat /L /P /A /R
```

DISKCOMP

Format

```
DISKCOMP [/H] [d: [d:]] [/1] [/8] [/V]
```

Explanation

DISKCOMP compares two entire diskettes track by track. This is useful, for example, to check that a DISKCOPY operation has worked correctly.

If you specify DISKCOMP with no drives, DISKCOMP assumes the current drive. If you do not specify two drives, for example if you type:

```
A>DISKCOMP a:
```

DISKCOMP prompts you to swap diskettes as necessary.

You can also use DISKCOMP to verify that an entire diskette can be read, by specifying the /V switch.

Command switches

/1	Compares only the first side of the diskettes, even if the diskettes are double-sided.
/8	Compares only the first eight sectors per track of a diskette, even if the first diskette contains more sectors per track.
/V	Verifies that the whole disk can be read.

If a mismatch is found, DISKCOMP displays a “Compare” error message showing the track and side number where the mismatch was discovered.

Remarks

If you copy all the files from one diskette to another with the COPY (rather than DISKCOPY command), DISKCOMP will probably return an error, even if the copying is done correctly. This is because COPY may place files on a destination diskette in a different order from the source diskette. (The way to check that a COPY operation has worked is to use the /V switch.)

DISKCOMP does not work with networked drives or with drives affected by a ASSIGN, JOIN or SUBST command.

The first diskette must be the same type as the second diskette (for example 360 Kbytes, 1.2 Mbytes), or you get an error message.

Examples

The following command compares the first diskette in drive A: with a second diskette in drive A: (you are prompted to change diskettes where necessary):

```
A>DISKCOMP
```

The following command compares the diskette in drive A: with the diskette in drive B:

```
A>DISKCOMP a: b:
```

The following command verifies a diskette in drive B:

```
A>DISKCOMP b: /V
```

DISKCOPY

Format

```
DISKCOPY [/H] [source_drive] [destination_drive] [/1]
```

Explanation

The DISKCOPY command copies entire diskettes of the same format. If the destination disk has a different format, or is unformatted, DISKCOPY will format it to match the source disk (where this is possible). If you want to copy files between diskettes with incompatible formats, use the XCOPY command.

WARNING DISKCOPY only works with diskettes, not with a hard or virtual disk. DISKCOPY erases anything that exists already on the destination diskette.

If you omit the *destination_drive*, DISKCOPY copies to the default drive, not the *source_drive*. If the source and destination drives are identical, DISKCOPY prompts you when you need to swap disks. If you omit both drives, DISKCOPY copies to the default drive.

Command switch

/1	Causes DISKCOPY to copy only side 0 of a diskette, enabling you to copy single-sided diskettes.
----	---

You need to know your drive and diskette types before you attempt a DISKCOPY. Certain disk types cannot be copied in certain drive types. The table that follows lists the disk and drive types supported by DISKCOPY.

Note

If you want to copy a *complete structure* of directories and files from one type of disk to another type of disk, you can use the XCOPY command. (This will not include a volume label.)

Table 11-6 Disk and Drive Types Supported by DISKCOPY

Disk size	Type	Tracks	Capacity
5¼"	Single sided, double density	40	180 Kbytes
5¼"	Double sided, double density	40	360 Kbytes
5¼"	Double sided, high density	80	1.2 Mbytes
3½"	Double sided, double density	80	720 Kbytes
3½"	Double sided, high density	80	1.44 Mbytes

Remarks

Note the following:

- ☐ A 180 Kbyte drive can copy 160 and 180 Kbyte diskettes.
- ☐ A 360 Kbyte drive can copy 160, 180, 320 and 360 Kbyte diskettes.
- ☐ A 1.2 Mbyte drive can copy 1.2 Mbyte diskettes. It can also copy 160, 180, 320 and 360 Kbyte diskettes, but you may not be able to read these diskettes in a 180 or 360 Kbyte drive.
- ☐ A 720 Kbyte drive can only copy 720 Kbyte diskettes.
- ☐ A 1.44 Mbyte drive can copy 720 Kbytes and 1.44 Mbyte diskettes.

Example

If you enter the command:

A>DISKCOPY a: a:

DISKCOPY prompts you to insert the source diskette. When the information on the source diskette is copied into memory, you are prompted to insert the destination diskette. The information in memory is then copied onto the destination diskette. This may be repeated several times depending on the size of the disk.

ERAQ

Format

ERAQ [/H] wildspec [/S]

Explanation

ERAQ erases files, having first queried whether you wish to continue.

ERAQ is most useful when used with the wildcard characters * and ? to erase an entire class of files. Instead of erasing every file that matches the file specification, ERAQ lists the specified files individually, followed by a ? prompt, to which you respond either with a Y for Yes (erase the file) or with an N for No (do not erase the file).

Command switches

/S	Includes system files.
----	------------------------

Example

This example gives you the choice to delete each of the **.bak** files in the current directory.

A>ERAQ *.bak

ERASE (ERA)

Format

ERASE [/H] wildspec [/C] [/S]

Explanation

ERASE and ERA have the same format, and each deletes the specified file or files. The file specification can identify a particular file, or it can indicate a group of files through the wildcard characters * and ?.

After you erase a file, you cannot retrieve it, so you should be very careful when you use wildcards. Generally, ERASE does NOT query you before erasing files. However, it does query you when you use the asterisk (*) wildcard for both the filename and the extension (*.*). For *.* it displays the following query:

Are you sure (Y/N)?

Type Y to erase all files on the (directory); type N to cancel the command.

Command switches

/C	Prompts you to confirm deletion.
/S	Includes system files.

Example

This command erases all the **.bak** files on the current directory, without querying you.

A>ERA *.bak

EXE2BIN

Format

```
EXE2BIN [/H] filespec1 [filespec2] [/Sxxxx]
```

Explanation

EXE2BIN is a utility, mainly for use by program developers, that converts **.exe** type executable program files produced by a DOS Linker into a **.bin** binary image or **.com** executable file.

Filespec1 represents the path to, and the name of, the source **.exe** file. If no extension is specified a default of **.exe** is used.

Filespec2 represents the path to, and the name of, the destination file to be produced. If *filespec2* is not specified, the file given in *filespec1* is used with an extension of **.com** or **.bin**. If *filespec2* is specified but without an extension, the source file is used with a **.com** or **.bin** extension depending on the contents of the source **.exe** header.

Files with the extension **.exe** produced by a DOS Linker have a special header which includes a relocation table. The header provides DOS with information on the size, memory requirements and entry point of a program as well as the locations of absolute segment address references. Files with the extension **.com** do not contain this special header as they are absolute memory images of the program to be executed.

Whether the file produced is a **.bin** or **.com** type file depends on what is set in the initial IP (instruction pointer) field in the source **.exe** file header. If the IP field contains a value of 100H, a **.com** file is produced. If the IP field contains a value of 0, a **.bin** file is the result. This is not applicable if the destination filename extension is specified.

A **.exe** program must have the following characteristics before it can be converted to a **.com** file:

- ☐ The program must have a maximum size once loaded of 65277 bytes.
 - ☐ The program must have its entry point offset 100H within its segment.
 - ☐ The program must not have a stack segment declared.
-

- ❑ The program must have no absolute segment address references which require segment fixups stored in the **.exe** header. (Segment fixups are needed if the program contains instructions requiring segment relocation, which is usually done by the DOS loader using the **.exe** file's relocation table.)

For an **.exe** program to be converted to a **.bin** file it must:

- ❑ have its entry point at offset 0 in its memory usage.
- ❑ not have a stack segment declared.

Command switch

<i>/Sxxxx</i>	<i>/S</i> followed by four hexadecimal digits gives the base segment value to be used if any segment fixups are required for the .bin file. If fixups are needed, but this parameter is omitted, you are prompted to enter a value.
---------------	--

Examples

This command converts the **.exe** file **data.exe** to the file **data.bin**:

```
C>EXE2BIN data
```

The following command converts the file **utility.tst** (which includes a valid **.exe** header and fulfils the requirements for conversion to a **.com** file) to the file **result.fin** (which will be a valid executable file):

```
C>EXE2BIN utility.tst result.fin
```

The following command converts the file **utility.tst** (which includes a valid **.exe** header and fulfils the requirements for conversion to a **.bin** file) to the file **abinfil.fin** (which will be a valid **.bin** file) using a base segment fixup value of 1234 hexadecimal:

```
C>EXE2BIN utility.tst abinfil.fin /S1234
```

EXIT

Format

EXIT [/H]

Explanation

This command returns you to another program. Some programs allow you to leave them, work with operating system commands, and then return to what you were doing before.

EXIT is the command you use to return from the system prompt to the program.

EXIT also stops a secondary **command.com** if one is loaded, (see the entry for **COMMAND**).

FASTOPEN

Format

FASTOPEN [/H]

Explanation

FASTOPEN reduces the time taken to access files on a hard disk which are being repeatedly accessed. It does this by remembering the locations of the files on disk, so that the operating system does not have to navigate along the path every time the file is opened.

A special table of these filenames is set up in memory. You can set the size of this table by inserting a FASTOPEN statement in your **cconfig.sys** file. This can be done at installation using INSTALL, later using the SETUP program or by editing the **cconfig.sys** file with an ASCII editor.

You cannot change the FASTOPEN setting by entering the command at the command line prompt. It is present as a command only because certain applications require it in order to operate correctly.

See Chapter 16, "Customizing your system," for more information.



FC

Format

```
FC [/H] [@]filespec1 filespec2 [/A] [/B] [/C] [/Gn]
[/L] [/Mn] [/W]
```

Explanation

FC is a file compare utility. Use it to do a comparison of two ASCII or binary files and display the differences.

With ASCII files the comparison is line-by-line. The ensuing display tells you where to insert, delete or replace lines in *filespec1* to make it the same as *filespec2*.

With binary files the comparison is byte-for-byte. There is no attempt made to suggest amendments to *filespec1* to match it with *filespec2*.

In the case of **.exe**, **.com**, **.sys**, **.obj**, **.bin** and **.lib** files the default comparison is byte-for-byte. With other files the default is line-by-line.

Command switches

/A	Abbreviates output during an ASCII comparison.
/B	Forces a binary byte-for-byte comparison.
/C	Ignores case during an ASCII comparison.
/Gn	Sets the number of lines that must match before resynchronization occurs. The default is five.
/L	Forces an ASCII line-by-line comparison.
/Mn	Sets the maximum number of mismatches that can occur during a binary comparison. The default is 20. For an unlimited number enter 0.
/W	Ignores white space (spaces and tabs) during an ASCII comparison.

Examples

The following command compares two files in ASCII mode:

```
C>FC oldfile newfile
```

The following command compares two files in binary mode:

```
C>FC oldfile newfile /B
```

FDISK

Format

FDISK [/H] [/D]

Explanation

FDISK is a menu-driven utility which is used to divide a hard disk into partitions.

Before you can use a hard disk you need to partition it. This means to define *sections* on the disk. The maximum size section or partition you can have is 512 Mbyte (you can choose to have smaller partitions if you wish). If you are preparing a hard disk for the first time you will need to prepare a special DOS partition in which to install the operating system. This is called the *primary* partition; see "Create DOS primary partition" later in this command explanation.

Using FDISK you can create one primary DOS partition and one extended DOS partition, which may contain as many logical disk drives as you require. You can then pass information between DOS partitions on a hard disk as if between different drives.

For example, if the disk is divided into a primary partition and an extended partition containing two logical drives, you can address the partitions as follows:

- ☐ C: is the primary partition.
- ☐ D: is the first logical drive on the extended partition.
- ☐ E: is the second logical drive on the extended partition.

Command switch

/D	Allows the deletion of an existing non-DOS partition. If you do not specify /D on the command line, you are not allowed to delete a non-DOS partition. Use with care.
----	---

When you type FDISK, the current partition status of the hard disk is displayed, followed by a Main Menu of options.

The different items of status information shown are:

Field	Meaning
No.	Each partition in the disk is numbered in the order in which partitions appear on the disk, starting at 1.
Drive	The drive letter assigned to the hard disk drive, starting with C.
Start	The starting cylinder number for the partition.
End	The ending cylinder number for the partition.
MB	The size of the partition in Mbytes.
Status	Either A or N. A indicates that the partition is <i>active</i> , that is bootable (your computer attempts to boot from this partition if started from the hard disk). N means <i>not active</i> .
Type	The format of the partition. This can be DOS (for example this system or another DOS operating system) or non-DOS.

The options you can select from the Main Menu are described below.

Create DOS partition

Select this option if you are:

- ☐ Preparing a hard disk for the first time.
- ☐ Adding a new partition to a hard disk.

FDISK warns you if there is not enough space to create a new partition or if all partitions are already in use.

When you select this option, you are asked which type of partition you want to create:

Select options:

- 1) Create DOS primary partition
- 2) Create DOS extended partition
- 3) Create logical drive in extended partition

Enter desired option: (ESC = exit) [?]

Remarks

- ☐ You can create a primary partition only when you format a hard disk for the first time and no primary partition already exists on the disk.
- ☐ You can create an extended partition only if a primary partition already exists on the disk.
- ☐ You can create a logical drive only if you have an extended partition.
- ☐ A warning message is displayed if you make an invalid selection.

Create DOS primary partition

When you select the **Create DOS primary partition** option you are asked if you want to use the largest available space on the hard disk (this example is for a 40.2 Mbyte drive):

Use cylinders 1 - 975 for DOS (40.2 MB) (Y/N)?

If you want to use all the available space, type **Y** to accept the default selection and press **Enter**.

If you do not want to use all the available space (for example if you want to have an additional operating system), type **N** and press **Enter**.

Note

If you plan to add another operating system to the hard disk, check the size requirements for that system *before* you allocate space to DOS.

You are then prompted for the starting and ending cylinders:

[Enter] starting cylinder:

[Enter] ending cylinder:

If FDISK encounters a defective track on the system area of the disk, it does not create a partition. FDISK prompts you to try again with another starting cylinder number. Similarly, if there is insufficient space at a cylinder, FDISK prompts you to try another cylinder number.

You are prompted to:

[Enter] disk label:

You can give the partition a label of up to eleven characters.

Once you have chosen your partition size you are returned to the Main Menu. The screen displays details of the new

partition you have created. Press **Esc** to leave the FDISK program.

The partitions you create contain no program or data files so you must load the files you want into the appropriate partition. For example, you may want to load the operating system into a partition so that you can boot the system from the hard disk. If you are using FDISK as part of the system installation procedure, the INSTALL program automatically does this for you. So when you have chosen your partition size and you are asked to reboot, simply leave your distribution diskette in drive A: to restart the INSTALL program.

You can use INSTALL to load the operating system into your partition or you can switch to drive A: and use the SYS command. Then use option 3, "Select bootable partition," to make the partition active.

Create DOS extended partition

Select the Create DOS extended partition option if you already have a DOS primary partition and want to add another partition to your disk. You are asked to specify the starting and ending cylinders for the partition, in the same way as creating a DOS primary partition. You are then returned to the Main Menu and the screen displays details of the partition you have just created.

Note

You cannot use an extended partition until you have created one or more logical drives within it. The extended partition is not assigned a drive letter of its own.

Create logical drive in extended partition

Choose the Create logical drive in extended partition option to create a logical drive in an existing extended partition. When you select it the screen displays the current status of the extended partition. You are then asked to specify the starting and ending cylinders for the logical drive.

All logical drives must be contained completely within the extended partition. When the drive has been created you are offered the choice of creating another logical drive or returning to the Main Menu. Press **Esc** to return to the Main Menu.

Delete DOS partition

Select this option to delete a current partition.

WARNING When a partition is deleted, you can no longer access data in that partition, so use this option with care. Back up first all files held in the partition (either on diskettes or in another partition on the hard disk).

When you select this option, you are asked which type of partition you wish to delete:

Select option:

- 1) Delete DOS primary partition
- 2) Delete DOS extended partition
- 3) Delete logical drive in extended partition

Enter desired option: (ESC = exit) [?]

Note

You can only choose option 2 to delete an extended partition if the partition does not contain any logical drives. If the partition you want to delete contains logical drives you must delete them first.

Delete DOS
primary/extended
partition

If you select Delete DOS primary partition or Delete DOS extended partition, you are prompted to:

Enter number of partition to delete (ESC=exit)

Type in the number of the partition you want to delete and press **Enter**, you will see:

Do you really want to erase all data (Y/N)?...

If you are ready to delete the partition, type Y and press **Enter**.

Delete logical drive in
extended partition

If you select the Delete logical drive in extended partition option the screen displays the logical drives in the extended partition before prompting you to select a drive to delete.

Select bootable partition

Use this option to select which partition on a hard disk is to be *active*. This is the partition containing the operating system and files you want to access when you switch on your computer. If you designate a particular partition as active, your computer

searches this partition if the system is booted from the hard disk. Only one partition can be active at a time.

When you select this option, you are prompted to:

Select partition to make bootable.....

Type in the number of the partition you want to make active and press **Enter**.

Note

If your hard disk only contains partitions for this operating system, then only the primary partition can be made active. If you try to select an extended partition you will get an error message.

This partition now has an active status (A). The partition that was previously active is now non-active (status N).

Your computer will always try to load an operating system from drive A:. If there is no diskette in drive A: or if drive A: is not working, the computer tries to load an operating system from the active partition on the hard disk.

Display logical drives in extended partition

You only see this option in the Main Menu if you have an extended partition on your disk. If you select it the screen displays the current status of the logical drives in your extended partition, for example:

Logical drives in extended partition (8.5Mb, 205 cylinders):

No	Drive	Start	End	MB	Type (i.e no status)
1	D	771	975	8.4	DOS 2.0

Select alternate fixed disk

You only see this option in the Main Menu if you have multiple hard disks on your computer. Select it to switch from one disk to another. The screen displays the current partition status of the disk as follows:

Partitions on 2nd hard disk (20.3 MB, 614 cylinders):

No	Drive	Start	End	MB	Status	Type
1	F:	1	304	10.1	A	DOS 3.0
2	-	305	613	10.2	N	(other)

FILELINK

Format

```
FILELINK [/H] command [[@]wildspec1] [wildspec2]  
[comparm] [/A] [/D:date] [/H] [/M] [/P] [/R] [/S]  
[/U] [/X]
```

Explanation

FILELINK is a utility used to transfer files between two computers connected by their serial communication ports. This can be useful when you are not able to transfer files via diskettes, perhaps because one computer has a diskette drive of a different type from the other, or does not have a diskette drive at all.

You can also use it to look at a directory of files on another computer. FILELINK must be running on both computers in order to transfer files or produce a directory listing; a useful feature of the utility is that you can install it from one computer to the other using the serial communications port.

FILELINK operates in two different modes: *master* and *slave*. When FILELINK is running in slave mode, it only acts in response to requests received from the other computer running in master mode across the serial communication line, usually to send or receive files.

Note

To avoid confusion when we discuss FILELINK, the computer you are typing a command *on* is referred to as the *master* computer, and the computer connected *to* it via the serial cable is the *slave* computer. All FILELINK commands are entered at the command line on the *master* computer.

Command options

<i>command</i>	FILELINK command (the valid abbreviation is shown in brackets): DIRECTORY (DIR) DUPLICATE (DUP) QUIT (QUI) RECEIVE (REC) SETUP (SET) SLAVE (SLA) TRANSMIT (TRA)
<i>@wildspec1</i>	The drive, path and name of the file to be copied. Files are renamed if a destination filename is specified. If <i>wildspec1</i> is preceded by @, it specifies the drive, path, and name of a file that contains a list of the files to be copied.
<i>wildspec2</i>	The destination drive, path, and name of the file.
<i>comparm</i>	The communications port and baud rate to be used, in the form COM#:baudrate, where # is the COM port number (for example 1 or 2), and baudrate can be for example, 9600 or 19200 depending on your own computer.

Cabling requirements

Before using FILELINK a suitable cable must connect your two computers.

The type of cable you need is an RS-232C Null Modem cable. The cable must have connectors that are appropriate to the computers to be linked. Normally your computers will have either 9-pin, or 25-pin D-type male connectors for their serial ports. Therefore the cable to connect them must have 9 or 25-pin (as appropriate) D-type female connectors, see the table that follows. (The same cable can also be used as a serial cable.)

Table 11-7 Serial Cable Pin Connections

25 pin to 25 pin		9 pin to 9 pin	
7	7 Ground	5	5 Ground
2	2 Transmit data	3	3 Transmit data
3	3 Receive data	2	2 Receive data
4	4 Request to send	7	7 Request to send
5	5 Clear to send	8	8 Clear to send
6	6 Data set ready	6	6 Data set ready
20	20 Data terminal ready	4	4 Data terminal ready

9 pin to 25 pin	
5	7 Ground
3	2 Transmit data
2	3 Receive data
7	4 Request to send
8	5 Clear to send
6	6 Data set ready
4	20 Data terminal ready

Baud rates

In a serial communication link, information is transferred one bit after another. The number of bits transferred per second is referred to as the *baud* rate. Due to various standard serial transfer protocols, only 8 bits of data are transferred for every 10 bits of information. So at 9600 baud, up to 960 bytes can be transferred every second. The higher the baud rate, the quicker your files are transferred. FILELINK supports up to 115,200 baud, although not every computer can work at this speed. You may want to start working with FILELINK with a baud rate of 9600 which most computers should accept. Then when you have successfully transferred files using FILELINK, and you know

that your serial ports and cable are correct, you can experiment with higher baud rates.

Whichever baud rate you use, it *must* be the same on each computer.

There are various standard baud rates available; they are listed in the following table with their valid abbreviations.

Table 11-8 Standard Baud Rates

Abbreviation	Baud rate
115	115, 200
56	56, 700
38	38, 400
19	19, 200
96	9, 600
48	4, 800
24	2, 400
12	1, 200
60	600
30	300
15	150
11	110

Setting the default
communications port
(SETUP)

You can set a default communications port to be used when FILELINK is invoked without a communications port specified in the command line. To do this use the FILELINK command SETUP in the form:

```
FILELINK SETUP comparm
```

where *comparm* is the communications port and maximum baud rate to be used.

If you do not set up the default communications port, you will have to remember to specify it each time you issue a FILELINK command.

Typing FILELINK SETUP without any parameters displays the current setup.

Examples

The following command sets a default communications port on the *master* computer to COM1 and the maximum baud rate to 9600 baud:

```
C>FILELINK SETUP com1: 9600
```

The following command displays the current setup:

```
C>FILELINK SETUP
```

```
Current FileLink setup is for: com1:9600
```

Installing FILELINK on
a slave computer
(DUPLICATE)

To install FILELINK on a slave computer via a serial link, use the FILELINK command DUPLICATE in the form:

```
FILELINK DUPLICATE [comparm]
```

where:

comparm is the communications port to be used on the master computer. If it is not specified, the default communications port is used.

You are prompted to carry out various operations on the slave computer to install FILELINK.

Examples

The following command copies the FILELINK program via the serial communications port to a slave computer. As no *comparm* is supplied, the default set by the FILELINK SETUP command is used.

```
C>FILELINK DUP
```

Putting a computer into
slave mode (SLAVE)

To put a computer into slave mode, use the FILELINK command SLAVE in the form:

```
FILELINK SLAVE [comparm] [/X]
```

Files cannot be transferred to or from a computer until the FILELINK SLAVE command is issued on the slave computer. If you do not specify a communications port, the default communications port is used. The current directory and drive when slave mode is entered becomes the default for files sent to, or received from, the slave computer.

The valid option you can use with this command is:

Command switch

/X	If you specify /X on the command line no files are overwritten on the slave computer.
----	---

Example

The following command puts a computer into slave mode. As no comparm is given, the default is used as set by the SETUP command:

```
C:\JESS>FILELINK SLA
```

Note that as SLAVE mode was entered when the current drive was C: and the current directory was **jess**, these will be the default drive and directory for the slave computer.

To quit FILELINK running in slave mode, press **Ctrl+C**, see also “Quitting FILELINK on a slave computer.”

To send a file or files from the master computer to the slave computer, use the FILELINK TRANSMIT command in the form.

Sending files to a slave computer (TRANSMIT)

```
FILELINK TRANSMIT [@]wildspec1 [wildspec2] [comparm]
[/A] [/D:date] [/H] [/M] [/P] [/R] [/S] [/U]
```

where:

wildspec1 is the drive, path and name of the file to be sent from the master computer.

wildspec2 is the destination filespec on the slave computer.

Before you can send files, the slave computer must be running in slave mode, see *Putting a computer into slave mode*.

The valid switches you can use with this command are:

Command switches

/A	Copy only files with the archive attribute.
/D:date	Only copy files modified since a specified date.
/H	Copy files with hidden or system attributes.

Command switches (continued)

/M	Copy only those files that have their archive attribute set and either do not exist on the destination or exist but with their archive attribute reset. When you specify the /M switch, you ask FILELINK to copy only those files that you have modified on the source directory, but have not modified on the destination directory. This makes use of the fact that the operating system automatically sets the archive attribute on any file that gets created or changed. You can use this switch to back up files regularly from one computer to another, as shown in the examples below.
/P	Prompt before copying each file.
/R	Overwrite read-only files on the destination.
/S	Copy files in subdirectories.
/U	Copy only files that do not exist on the destination, or that exist but are older versions, that is, have an earlier date stamp.

Examples

The following examples assume that the FILELINK SLAVE command was issued first on the slave computer, as in the example under "Putting a computer into slave mode."

If you have several documents with the file extension **.doc**, and you occasionally alter one or two of them, when you back them up you can copy them to the slave computer using the command:

```
C> FILELINK TRA *.doc
```

However because you know that most of the files are already on the slave computer from the last time you backed up, it takes less time if you use the /M switch, then only those files you have changed since the last backup are copied; the command you use is:

```
C> FILELINK TRA *.doc /M
```

Getting files from a
slave computer
(RECEIVE)

To receive files from a slave computer (which must be running in slave mode), use the FILELINK command RECEIVE in the form:

```
FILELINK RECEIVE [@]wildspec1 [wildspec2] [comparm]
[/A] [/D:date] [/H] [/M] [/P] [/R] [/S] [/U]
```

where:

wildspec1 is the drive, path and name of the file to be received from the slave computer. If *wildspec1* is preceded by @, it specifies the drive, path, and name of a file that contains a list of the files to be received.

wildspec2 is the destination wildspec on the master computer.

If you do not specify a communications port, the default communications port is used as set up by the FILELINK SETUP command. The valid options you can specify are the same as those described in "Sending files to a slave computer."

Examples

Note that the first command example assumes that the FILELINK SLAVE command was issued on the slave computer as in the example in "Putting a computer into slave mode."

The following command copies all files on the current drive and directory of the slave computer (**c:\jess**), onto the current drive and directory of the master computer (**c:\jane**):

```
C : \JANE> FILELINK RECEIVE *.*
```

The next command copies all files with the extension **.doc** in the current directory of the A: drive on the slave computer, to the directory **c:\letters** on the master computer:

```
C> FILELINK RECEIVE a:*.doc c:\letters
```

Displaying a list of files
on a slave computer
(DIRECTORY)

To display a directory of files on a slave computer (which must be running in slave mode), use the FILELINK command DIRECTORY in the form:

```
FILELINK DIRECTORY [filespec] [comparm] [/A]
[/D:date] [/H] [/P] [/S]
```

If you do not give a communications port, the default port is used (as set up by the FILELINK SETUP command).

The valid switches are:

Command switches

/A	Show only files with the archive attribute.
/D:date	Show only files modified since the specified date.
/H	Show files with hidden or system attributes.
/P	Pauses after each screenful of display.
/S	Show files in subdirectories.

Examples

The following command displays a directory of all files of the current directory of the current drive on the slave computer:

```
C>FILELINK DIR
```

The following command displays a directory of all files with the extension **.bat** in the current directory of the current drive on the slave computer:

```
C>FILELINK DIR *.bat
```

The following command displays a directory of all files with the extension **.bat** dated 1-1-90 or later, in the root directory of drive A: on the slave computer.

Quitting FILELINK on a
slave computer (QUIT)

To stop running FILELINK (in slave mode) on a slave computer, either press **Ctrl+C** at the slave computer's keyboard, or use the FILELINK command QUIT from the master computer in the form:

```
FILELINK QUIT [comparm]
```

where:

comparm is the communications port on the master computer, if one is not given the default is used as set up by the FILELINK SETUP command.

Example

The following command stops FILELINK running on a slave computer via the communications port com2:

```
C>FILELINK QUI COM2
```

FIND

Format

```
FIND [/H] [/B] [/C] [/F] [/N] [/S] [/U] [/V] "search string"  
[@]wildspec...
```

Explanation

FIND searches for a string of characters in a group of text files and displays the lines containing the *search string*. The *search string* must be enclosed in double quotes (""). If the string includes double quotes, you must include an extra pair (as in ""find this").

Multiple files may be specified on the command line using the wildcard characters ? and *.

If no files are specified FIND reads from standard input (usually the keyboard or a redirected file).

Command switches

/B	Changes display format of the list that is produced, so it shows a series of headings with the matches listed beneath.
/C	Shows only the total number of lines that contained the string.
/F	Shows only the names of those files that contain the "string".
/N	Displays line numbers.
/S	Searches files in subdirectories that are derived from the current directory.
/U	Case-sensitive search (for example "A" doesn't match "a").
/V	Displays the lines <i>not</i> containing the string.

Examples

The following command calculates the number of lines containing the word *sorry* in each of the three files **apology1.let**, **apology2.let** and **apology3.let**:

```
A>find /C "sorry" apology1.let apology2.let apology3.let
```

The following command displays all the subdirectories listed in the current directory. FIND acts as a filter, receiving input from the DIR command and displaying as output only those files that contain the string:

```
A>DIR | find "<DIR>"
```

FORMAT

Format

```
FORMAT [/H] d: [/1] [/4] [/8] [/B] [/F:nnnn] [/N:n]
[/S] [/T:nn] [/V]
```

Explanation

The **FORMAT** command formats diskettes into standard formats. (Use **FDISK** to format hard disks.) The default format for a diskette depends on the drive type, for example the default format for a 360 Kbyte diskette drive is a 360 Kbyte diskette. This default can be changed by using some of the **FORMAT** switches.

WARNING **FORMAT** will wipe everything off the diskette. Use it with great care.

FORMAT supports the disk and drive types shown in the table that follows.

Table 11-9 Disk and Drive Types Supported by **FORMAT**

Disk size	Type	Tracks	Capacity
5¼"	Single sided, double density	40	180 Kbytes
5¼"	Double sided, double density	40	360 Kbytes
5¼"	Double sided, high density	80	1.2 Mbytes
3½"	Double sided, double density	80	720 Kbytes
3½"	Double sided, high density	80	1.44 Mbytes

Command switches

- /1** Forces single sided formats on a double sided drive. (Only allowed on 5 1/4" drives.) To use /1 you must use either /4 or /F:360 first to instruct your diskette-drive that the diskette to be formatted is 360 Kbytes.
- /4** Forces 360 Kbytes format. Use this to format 360 Kbytes format diskettes in a 1.2 Mbyte format drive. Note that diskettes formatted this way cannot be reliably read by 360 Kbytes drives. This has the same effect as /N:9 /T:40.
- /8** Formats 9 sectors but uses only 8 (only allowed on 5 1/4" drives). The default is to format 9 sectors and use all 9. To use /8 you must use either /4 or /F:360 first to instruct your diskette-drive that the diskette to be formatted is 360 Kbytes.
- /B** Formats 9 sectors but uses only 8. Reserves space for DOS system files. (Only allowed on 5 1/4" drives.)
- /F:nnnn** The capacity of the drive to be formatted in Kbytes. Use this switch when the format required is other than the default for the drive.

Values can be:

<u>For 5 1/4" diskettes</u>	<u>For 3 1/2" diskettes</u>
160 (160 Kbytes)	720 (720 Kbytes)
180 (180 Kbytes)	1440 (1.44 Mbytes)
320 (320 Kbytes)	
360 (360 Kbytes)	
1200 (1.2 Mbytes)	

As an alternative to the /F:nnnn switch, you can use the /T: and /N: switches to specify the diskette type:

- /N:9** Forces 9 sectors per track. (Allowed on 3 1/2" 1.44 Mbyte drives or 5 1/4" 1.2 Mbyte drives in conjunction with /T:80 or /T:40.)
-
-

Command switches (continued)

/N:8	Forces 8 sectors per track (on 1.2 Mbyte drives in conjunction with /T:80).
/S	Copies operating system files onto the formatted disk, giving the names of files copied. This switch cannot be used with 160 and 180 Kbytes disks because of their limited capacity. The system first looks in the root directory of the disk in the current default drive for the system files and uses COMSPEC to find command.com . If they cannot be found, you are prompted to insert a diskette containing the system files in the FORMAT drive. After reading the system files, you are prompted to remove the diskette and insert the diskette to be formatted.
/T:80	Sets the number of tracks to 80. (Used in conjunction with /N:9.)
/T:40	Forces a 40 track diskette to be formatted. (Only allowed on 5 ¼" 1.2 Mbyte drives.)
/V	Specify the volume label to use on the disk. The volume label can be up to 11 characters in length.

Remarks

FORMAT does not recognize drive assignments made with the ASSIGN command.

You cannot format drives over a network.

Do not use FORMAT with a SUBST command.

Examples

To format a diskette:

- 1** On a hard disk system with one diskette drive, insert a blank diskette into drive A: and type:

C>FORMAT a:

You are prompted:

Insert diskette to format in drive A:
and strike a key when ready

- 2** Press **Enter**. A message appears showing the progress of the formatting. When formatting is complete, you are prompted:

Format another diskette (Y/N)?

- 3** Press **Y** to format another diskette and insert a blank diskette into drive A: and follow the procedure through again. Otherwise type **N**.

On a twin-diskette system, with your system diskette in drive A:, type:

A>FORMAT b:

and place the diskette to be formatted in drive B:.

To format a diskette at 360 Kbytes with the diskette in a 1.2 Mbyte drive:

A>FORMAT a: /F:360

IDLE

Format

IDLE [/H] [=] ON|OFF

Explanation

IDLE displays or sets idle detection.

Many interactive DOS programs are written without consideration for the effect of running on a multiuser system. These programs look repeatedly to see if a key has been pressed and are effectively “idle” while this is happening, until, that is, a key does get pressed. This wastes processor time that could be used to improve the performance of other users and programs running on the system.

Typed with no parameters, IDLE displays the current idle-detection state.

To set idle detection to ON, type:

IDLE ON

To set idle detection to OFF, type:

IDLE OFF

The = character is optional.

When IDLE is set to ON, the operating system detects that an application is idle and allows other programs to use the saved processor time until the application needs to process such events as a mouse movement or keyboard entry.

The IDLE statement in **cconfig.sys** is used to enable or disable idle detection globally. By default, idle detection is enabled (IDLE ON).

Note

Entering an IDLE command at the command line *only* affects the current DOS session.

It is more usual to enable and disable idle detection for each application using the PIFED command. PIFED settings apply to an application in all DOS sessions and override the current settings for an individual session.

KEYB

Format

KEYB [/H] [+|-]

Explanation

KEYB displays a message indicating which national keyboard is currently loaded.

The national keyboard is chosen during installation, or by an INSTALL statement in **cconfig.sys** (see Chapter 16, “Customizing your system”).

The following table shows the national versions supported, and their associated two-character codes.

Table 11-10 National Keyboards

Country	Code
Belgium	BE
Canada (French)	CF
Denmark	DK
Finland	SU
France	FR
Germany	GR
Italy	IT
Latin America	LA
Netherlands	NL
Norway	NO
Portugal	PO
Spain	SP
Sweden	SV
Swiss (French)	SF
Swiss (German)	SG
United Kingdom	UK
United States	US

The following command switches are available:

Command switches

- | | |
|---|--|
| + | Forces the keyboard program to assume an enhanced(101/102/104 key) keyboard.

KEYB usually recognizes enhanced keyboards automatically. However, if KEYB does not recognize your enhanced keyboard, and you find that certain keys do not match the characters displayed, try re-issuing the KEYB command with the + option. |
| - | Forces the keyboard program to assume a standard (83/84 key) keyboard. |
-

Remarks

If you wish to change keyboards while the program is running, you can switch back to the default (USA) keyboard by holding down the **Ctrl** and **Alt** keys and pressing **F1**. To go back to your chosen national keyboard, type **Ctrl+Alt** and then **F2**.



KEYS

Format

KEYS [/H] [filespec] [/B]

Explanation

Allows you to set the DOS session switching keys while the operating system is running.

The default DOS session switching keys are defined in the `INSTALL` and `SETUP` programs, but using `KEYS` each user can change these key strokes for their terminal only. These will apply until they are reset with another `KEYS` command or until the operating system is rebooted.

There are three ways to change between DOS sessions. `KEYS` allows you to change one, two or all three of them:

- ❑ **Key+Num** (where *Num* is a valid DOS session number) takes you straight to a DOS session. For example, you could elect to use **Ctrl+2** to move to the second DOS session.
Instead of one key, you can elect to use a combination of control keys, for example, **Ctrl+Alt+Num**.
- ❑ **Key+Esc** moves you through the DOS sessions in sequence.
- ❑ **Key+Esc** displays a window showing the process running in each DOS session. You then use the cursor to indicate which DOS session you want to move to.

Note

If you use both the second and third methods, be sure to select different key(s) for them.

When you enter the `KEYS` command without the *filespec* parameter, a menu is displayed. Follow the instructions on the screen.

The following command switch is available:

Command switch

/B	Forces menus to be displayed in black and white. (Some mono displays attempt to simulate color displays with shades of grey, and can prove difficult to see.)
-----------	---

Before exiting from the KEYS command, you are invited to save your settings in a file. You can select a *filespec* of your choice, or accept the default of **keys.ssk**. Once you have such a file, you can load your chosen settings at any time by giving the *filespec* as parameter to the KEYS command, for example:

C>KEYS keys.ssk

By entering such a command in your **autoexec.bat** file, your chosen switch keys will be available on your terminal each time you start up your system.

LABEL

Format

```
LABEL [/H] [d:] [label]
```

Explanation

The LABEL command allows you to create, modify or delete the volume label on a disk or diskette. A volume label is a name given to a disk to identify it and is displayed when you perform a DIR, TREE or CHKDSK command. Note you can also use the VOL command to find out if a disk or diskette has a volume label.

Adding a label

To create a volume label, use the form:

```
LABEL label
```

where *label* is a string of up to 11 characters. You can also use spaces, but not tabs. As with naming files and directories, do not include the following characters:

```
<> = , ; : . * ? [ ] ( ) / \ + ! ~
```

If no drive is specified in the command line, the default drive is used. You are prompted to type a label if you do not specify one at the command line.

LABEL does not work with drives affected by a SUBST command. You cannot use LABEL on a networked drive.

WARNING

When you enter a new label it overwrites an existing label without giving any warning.

Deleting a label

To delete a label, type LABEL and the disk drive from which you want the label deleted.

To delete a volume label on drive B:, type:

```
A>LABEL b:
```

You get a message similar to this:

```
Volume in drive B: is WORKDISK
Enter Volume label (0 to 11 characters):
```

Press **Enter** and you see this message:

```
Delete current volume label (Y/N)?
```

Type **Y** to delete the label.

LIMSIZE

Format

LIMSIZE [/H] [nnnn]

Explanation

LIMSIZE limits the amount of expanded memory available to an application. (To limit the size of conventional memory, see the MEMSIZE command.)

Each DOS session has a default LIMSIZE value defined during installation (the INSTALL program) or by using the SETUP program. Initially, the default size is 1024 Kbyte, which should be suitable for most purposes.

You will probably only want to alter LIMSIZE if you want to:

- ☐ Make room for a DOS session to run a program with unusually large memory requirements.
- ☐ Decrease the available memory space on other DOS sessions so that a particular DOS session can have a very large memory allocation.

To change the LIMSIZE of a DOS session, bring that DOS session to the foreground, then type the command.

If you type LIMSIZE without an associated number, the LIMSIZE of the current DOS session is displayed.

Remarks

The LIMSIZE command only affects the current DOS session and only until the operating system is restarted.

You can only reduce the LIMSIZE value from that set in INSTALL or SETUP.

All LIMSIZE values are rounded up to the nearest 16 Kbyte.

Example

The following command sets the LIMSIZE on the current window to 2000 Kbytes:

A>LIMSIZE 1990

LOCK

Format

LOCK [/H]

Explanation

LOCK temporarily locks the users terminal on a system with security enabled. This is useful if you need to leave your terminal for some time and do not want to stop any programs that you have running, but do not want other users to be able to use your terminal.

To use LOCK you must move to a DOS session that is not in use. Then type the LOCK command. Normally, a message (stored in the file **lock.txt**) is displayed and no further input is accepted from the keyboard until you type in your login password.

You can now leave the terminal. Any programs running in other DOS sessions continue to be processed in the normal way.

When you type in the login password, the keyboard is unlocked and you can carry on working.

Example

When you type the LOCK command, you see a message similar to:

C>LOCK

This station is locked for user ismith

To unlock please enter your login password

LOGOUT

Format

LOGOUT [/H]

Explanation

Use this command to log out, if you are using a system in which you have logged in. When you have logged out, the login screen is displayed ready for you or another user to log in.

Before you log out, check that you have stopped any programs running on all the DOS sessions you started. If any programs are still running, you will see a message displaying the DOS sessions on your terminal and showing which programs are still running:

The following programs are still running on this station:

DOS session	Program
-------------	---------

2	WS
---	----

3	DRAW
---	------

4	EDITOR
---	--------

Do you want to abort these programs and logout now (Y/N)?

Only answer Y if you are sure that these programs are not processing data. If you are not sure, answer N. Switch DOS sessions and check what is happening on each DOS session individually. Stop the programs before re-entering the LOGOUT command.

If you answer Y, all the programs will be aborted and you will be logged out. The Login screen is displayed.

MAIL

Format

MAIL [/H] [username] [<filespec] [/S]

Explanation

The MAIL utility lets you send and receive electronic mail.

Use the MAIL *username* format to send mail other users on your system. You identify other users by their login names.

You are invited to identify the subject of your mail message, and then type the text of your choice. When you have completed your message, press **Ctrl Z**. Your message is now sent to the *username* you have specified.

Use the < redirection symbol to take the contents of a *filespec* as the text of your mail message. In this case, the first line of the file (up to the first carriage return) is taken as the subject of the message. So, if you are sending a file containing text generated for other purposes, make sure you add a subject line to the beginning of the file.

Invoke MAIL without a *username* parameter to read any mail you have in your mailbox.

Command switch

/S	Gives the status of your mailbox, without automatically reading its contents.
----	---

MEMSIZE

Format

MEMSIZE [/H] [nnnn]

Explanation

MEMSIZE limits the amount of conventional memory available to an application. To limit the size of expanded (EMS) memory, see the LIMSIZE command.

Each DOS session has a default MEMSIZE set in the `Maximum memory per process` option in the `INSTALL` and `SETUP` programs. Initially, this is the maximum possible MEMSIZE, around 576 Kbyte on most computers, to provide sufficient room for all applications to work.

Some programs only take up the amount of memory they need, thus allowing a number of programs to run concurrently on different DOS sessions. Others, however, are designed to occupy all the memory that they can find, thus limiting the number of other programs that can be multitasked. You can restrict the memory such a program occupies by specifying a lower MEMSIZE value for the DOS session in which it is to run. Display the DOS session you will run the program in, then type the MEMSIZE command.

If you type MEMSIZE on its own, the current MEMSIZE of the current DOS session is displayed.

Remarks

Entering a MEMSIZE command at the command line only affects the *current* DOS session and only until the system is rebooted. It is more usual to set the amount of memory used by an application with the `PIFED` command. This overrides the current MEMSIZE setting while the application is running.

Example

The following command sets the MEMSIZE on the current DOS session to 300 Kbyte:

```
A>MEMSIZE 300
```

MKDIR (MD)

Format

```
MKDIR [/H] [d:]dirpath
```

Explanation

Use MKDIR to create subdirectories in the root directory or other subdirectories. You can abbreviate MKDIR to MD.

If you do not include a backslash (\) before the first subdirectory name in the directory path, MKDIR begins the path to your new directory from the current directory of the current or specified drive. If you do include the backslash, MKDIR begins the path at the root directory of the current or specified drive. For example if you are in the root directory and you want to create the subdirectory **hank** in the subdirectory **sales**, type:

```
A>MKDIR \sales\hank
```

If you are currently in the **sales** subdirectory and want to create the subdirectory **hank** from **sales**, type:

```
A>MKDIR hank
```

The path from the current directory leading to the new subdirectory must be entered and all the subdirectories in the path must exist. If a subdirectory is specified that does not exist, MKDIR displays the following message:

```
Path not found
```

The maximum number of characters in a path to directory (including backslashes) is 63.

When a directory is created, the operating system automatically adds the entries **.** and **..** that it uses as markers to locate and identify the directory; these entries cannot be deleted.

Use subdirectories to organize your data and command files into logical groups. You can have subdirectories with the same name, but they must have different paths. Similarly, files created in a directory must be unique within it, but you can create files with the same name in other directories.

Example

The following sequence of MKDIR commands:

```
MKDIR \sales
```

```
MKDIR\sales\hank
```

```
MKDIR\sales\mary
```

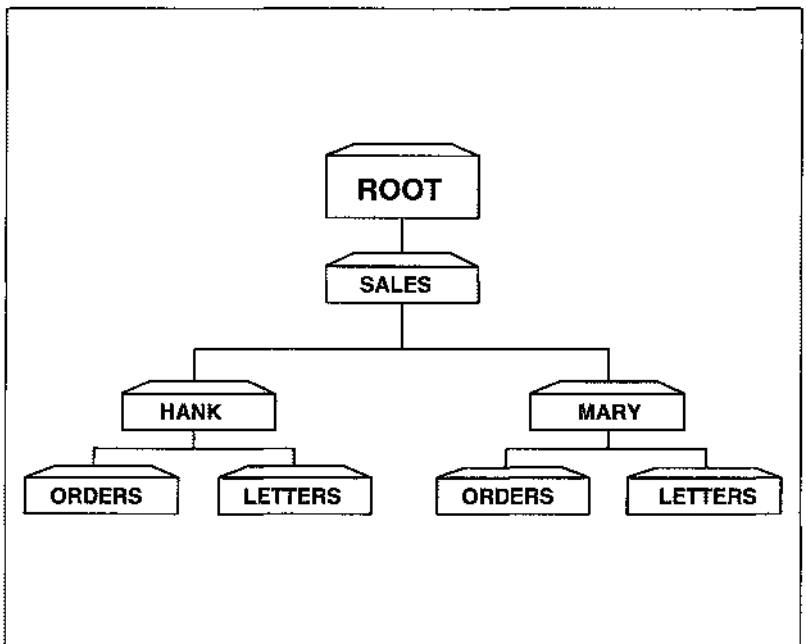
```
MKDIR\sales\mary\orders
```

```
MKDIR\sales\mary\letters
```

```
MKDIR\sales\hank\orders
```

```
MKDIR\sales\hank\letters
```

produces a hierarchical series of subdirectories that can be illustrated like this:



MODE

Format

MODE [/H]

Explanation

The **MODE** command, in general, allows you to change the configuration of various devices attached to your system. It has several formats each of which has its own separate explanation. These are as follows:

☐ **Set up screen display**

```
MODE displaymode[,lines]
MODE [displaymode],m[,T]
MODE CON[:] [LINES=lines][COL=cols][RATE=r DELAY=d]
```

☐ **Assign a printer**

```
MODE LPTn[:]=COMn
```

☐ **Set up a printer**

```
MODE LPTn[:] [n] [,m]
```

☐ **Set up a communications port**

```
MODE COMn[:]baudrate[, [parity], [databits],
[stopbits]
```

Set up screen display

Using **MODE** you can select the display type for your monitor, and change the current display. This can involve changing between 40 and 80 columns on the screen, or between monochrome and color. You can change the number of lines on the screen, or center the screen display. You can also alter your keyboard typematic rate (which is how quickly characters are echoed to the screen when a key is held down).

Setting the display type

To set the display type, use the following form:

```
MODE displaymode[,lines]
```

This format of MODE allows you to set the display type for a monitor.

The display types that can be selected are listed below. The mode you have selected remains until you type another MODE command or restart your system.

Option	Explanation
CO40	Enables the graphics adapter and sets the display width to 40 characters per line with a color display.
CO80	Enables the graphics adapter and sets the display width to 80 characters per line with a color display.
BW80	Sets the display width to 80 characters per line, and the display mode to black and white.
MONO	If a graphics adapter is enabled, switches the active display to the original monochrome display (if you have one fitted). This display always has a width of 80 characters per line.

If the graphics adapter is already enabled you can use the following commands:

Option	Explanation
40	Sets the display width to 40 characters per line.
80	Sets the display width to 80 characters per line.
MONO	Switches the active display to the original monochrome display (if you have one fitted). This display always has a width of 80 characters per line.
,lines	Sets the number of text rows (25, 43 or 50) on the screen.

Examples

The following command selects 80 columns by 43 lines:

```
C> MODE 80,43
```

On mono VGA, mode CO80 is often used to make an application believe that it has a color screen, or conversely is in mono mode. The following command when used on a system with a mono VGA screen, will select mono mode or color mode (with colors shown as gray scales):

```
C> MODE CO80
```

Aligning the display on
a CGA monitor

CGA or Color Graphics Monitors do not always display information properly. The MODE command can be used to center data so it is more readable. The format is:

```
MODE [displaymode],m[,T]
```

where:

Option	Explanation
displaymode	Sets the type of display you want to specify, see the list on the previous page.
,m	Shifts the display right or left; enter R for right, L for left.
,T	Causes a test pattern to be displayed so you can see how the screen will be aligned. You will be asked if the alignment is acceptable. If you answer N the display will be shifted again and you can continue this procedure until the display is correctly positioned.

Example

This command shifts the display one column to the right and then asks you if you can see all of the test pattern.

```
MODE ,R,T
```

Setting lines, columns
and the keyboard
typematic rate

To set the number of columns and lines on a display or the keyboard typematic rate (which is how quickly characters are echoed to the screen when a key is held down), use the MODE command in the form:

```
MODE CON[:] [LINES=lines] [COL=cols] [RATE=r DELAY=d]
```

where:

Option	Explanation
<code>LINES=lines</code>	Sets the number of lines on the display.
<code>COLS=cols</code>	Sets the number of columns on the display.
<code>RATE=r</code>	Sets the typematic interval time (that is the speed at which characters are repeated on the screen), in the range 1-32 where 1 is the <i>slowest</i> repeat.
<code>DELAY=d</code>	Specifies the auto-repeat start-delay time with the possible values of 1-4 (that is, if you hold down the F key on your keyboard, auto-repeat start-delay is how long it is before F is repeated). Note that 1 is the <i>longest</i> delay.

Note

To set the typematic rate you must specify the rate and delay at the same time.

Examples

The following command selects 80 columns by 43 lines on the display:

```
C>MODE CON: COL=80 LINES=43
```

This command sets the keyboard typematic rate to 32 (the shortest rate permitted), and the auto-repeat start-delay to 1 (the shortest delay allowed).

```
C>MODE CON: RATE=32 DELAY=1
```

Assign a printer

To assign a printer, use the form:

```
MODE LPTn[:]=COMn
```

With this form of MODE you can alter the standard assignments for your printer ports. It is provided for compatibility with other DOS operating systems.

You can access up to three printers from a DOS session. Usually information is sent to these printers through your DOS session's logical parallel ports, commonly labeled LPT1, LPT2, and LPT3. The MODE command enables you to send information to a serial printer through one of your computer's serial ports instead. (The serial ports are sometimes called COM ports or communications ports and are labeled COM1 and COM2). For example, if you have a program (such as a word processor) that normally sends output to port LPT1, you can use MODE to redirect information to a logical serial port like COM1 instead.

n represents the number of the port, for example LPT*n* could be replaced with LPT1. The colon between LPT*n* and =COM*n* is optional.

To return to the default settings, type the LPT*n*: without the COM*n*.

Note

This form of the MODE command overlaps with the functionality provided by the PRINTMAP command. Furthermore, the assignments made with the MODE command are to logical ports - they are mapped further according to the assignments set up with PRINTMAP. (See Chapter 13, "Printing", for more information about printing and the use of the PRINTMAP command.)

Example

To redirect output from LPT1 to COM1, type the following:

```
C>MODE LPT1:=COM1
```

Set up a printer

To set up a printer, use the form:

```
MODE LPTn[:][n][,m]
```

Use this form to set up a printer connected to a particular port. Your printer documentation should help indicate the values you need (for example for *n*).

Options	Explanation
LPT <i>n</i> :	The printer number, where <i>n</i> can usually be 1, 2, or 3.
<i>n</i>	The printer characters per line: 80 or 132. The default is 80. (80 character printers are often known as “narrow carriage”, and 132 character printers as “wide carriage”.)
<i>m</i>	The lines per inch for the printer: 6 or 8 (8 is the default).

Note

This form of the MODE command operates on the logical ports LPT*n*. The corresponding physical port is assigned by the PRINTMAP command.

Example

This command sends information to a 132-column printer at printer port LPT2:

```
C>MODE LPT2:132
```

Set up a communications port

To set up a communications port, use the form:

```
MODE COMn[:]baudrate[, [parity], [databits],  
[stopbits]]
```

This form enables you to set communications parameters for your serial (or communication) ports, for example the logical COM1 and COM2 ports for your DOS session. Use the PRINTMAP command to find out which physical ports the logical ports are on.

The command allows you to set baud rate, parity, data bits, and stop bits. Usually, you will find that you only need to set the baud rate; the default values for the other options may already be set up for your needs.

Options	Explanation
<i>baudrate</i>	Can be 110, 150, 300, 600, 1200, 2400, 4800, 9600, 19200, 38400, 56000 or 115200. You only need to type the first two digits of your selection.
<i>parity</i>	Can be N (none), O (odd) or E (even, which is the default).
<i>databits</i>	Can be 7 or 8 (7 is the default).
<i>stopbits</i>	Can be 1 (the normal default) or 2 (the default when the baud rate is set at 110).

Note

Always separate the different options in the command line with commas. If options are not required, the correct number of commas must still be inserted.

Examples

The following command sets the communications parameters (sometimes called protocol) on COM1 to 9600 baud. The defaults for the other values will be: even parity, 7 databits, 1 stopbit.

```
C> MODE COM1:96
```

The following command changes the protocol for the above command to odd parity:

```
C> MODE COM1:96,o
```

The following command sets the protocol on COM2 at 4800 baud and 8 databits:

```
C> MODE COM2:48,,8,
```

MORE

Format

source | MORE [/H]

Explanation

The MORE command is a filter that takes input from a standard input device (for example a redirected file or a piped command) and then displays it a screenful at a time. At the bottom of each screen (except the last one) the phrase:

Strike a key when ready . . .

is shown. Move to the next screen by pressing any key.

Example

Use MORE with the DIR command to display a large directory a screenful at a time. The output from DIR is piped to MORE and then displayed.

A> DIR c: | MORE

For more information on piping, see Chapter 12, “Redirecting input and output.”

NETDRIVE

Format

```
NETDRIVE [/H] [d: [/L | /R]]
```

Explanation

NETDRIVE is used to fool applications into believing that a local drive is remote. Some DOS LAN applications require this setting if they are to behave correctly on a multiuser system.

Options	Explanation
/L	Designates the drive <i>d:</i> as a local drive.
/R	Designates the drive <i>d:</i> as a remote drive.

NETDRIVE works on a per-DOS session basis.

If you invoke NETDRIVE with no parameters, the status of all the drives on the system is displayed. If you use the NETDRIVE *d:* format, the status of the specified drive is displayed.

Some disk editor-type applications will not work on a drive set to remote. If you need to use such an application, designate the drive as being a local drive first.

PASSWORD

Format

```
PASSWORD [/H] [[@]wildspec...]  
[/D|/G|/P|/R|/W:password] [/N] [/NG] [/NP] [/S]
```

Explanation

The **PASSWORD** command lets you perform two distinct passwording operations. On secure systems with user accounts in place, it lets you change your login password. On all systems it also lets you assign passwords to individual files (though on secure systems you are not likely to make use of this feature).

For full information on how to use login and file passwords, see Chapter 8, "Protecting your information."

To change your login password, enter the command without any parameters. To password individual files, use the command syntax as shown above. The **/H** switch gives help on both **PASSWORD** options.

Changing login passwords

Use the command to change your login password:

- ☐ as soon as you are given a user name and default password by the Superuser.
- ☐ at frequent intervals to make it harder for others to work out your password.

Passwords must be from three to 12 characters long. All the characters allowed in filenames can be used in passwords. Upper and lower case letters are taken to be the same.

When you enter the **PASSWORD** command without any parameters, you are prompted for your current password and then the new password. You must then repeat the new password to ensure that you did not mistype it. If the two do not match, you are returned to the system prompt and have to start again.

Note that when you enter your current and new passwords asterisks are displayed, for the sake of security.

If you forget your password, ask the Superuser to reset it.

Passwording files and paths

Use **PASSWORD** to assign passwords to files or paths when you are not using a secure system with user accounts enabled.

File passwords can be up to eight letters long.

If you want to check whether a file has been given a password and the restrictions on the file, type in the **PASSWORD** command and the full file specification. You see a display similar to:

```
C>PASSWORD C:\ALICE\MYFILE.TXT
rwd c:\alice\myfile.txt
```

In this example, no password has been assigned.

Command switches

<i>/D:password</i>	The password is required only for deleting or renaming the file. You do not need the password to read, modify or change the file's attributes. If you display the file password restrictions you see <code>rw-</code> .
<i>/G:password</i>	Sets the global default password. The system remembers this default password and tries it against each password-protected file you then try to access. This is useful if a group of files have the same password and you want to access the files without typing a password each time. The global default password remains set until it is removed, or until the computer is reset. It is also used before a non-password aware application is used with password-protected files.
<i>/N</i>	Removes any password protection. Note that you have to give the password first in order to remove it. <i>/N</i> can also be used in combination with the <i>/P</i> and <i>/G</i> options:
<i>/NG</i>	Removes the global default password.
<i>/NP</i>	Removes password protection from a directory.
<i>/P:password</i>	The password operates on subdirectories instead of files. This means each time you try to use the path to access a directory you are prompted for a password. Subdirectories are automatically given full read, write and delete protection when a password is applied.

Command switches (continued)

<i>/R:password</i>	The password is required for reading, copying, writing, deleting, renaming, and changing the file's attributes. If you display the file password restrictions you see --- before the filename.
<i>/S</i>	The password operates on files and subdirectories associated with the current directory.
<i>/W:password</i>	The file can be read and copied without giving a password, but the password is required to delete, rename or change the file's attributes. If you display the file password restrictions you see r--.

Note

If you are using a system on which system security has been enabled, assigning passwords places restrictions *in addition to* the owner, group and world access permissions. See the XATTRIB command.

Examples

The following example assigns the password SUZE to the file **eyesoff.let**, and ensures that the password must be given before the file can even be read:

```
A>PASSWORD eyesoff.let /R:suze
```

The following command protects all files with the **.bas** extension in the **\handsoff** directory and in all the associated subdirectories. The level of protection is set at W; the files can be read and copied but not renamed, deleted or altered. Note that you do not need to use the password to access the **\handsoff** directory or the subdirectories; you only need the password with ***.bas** files:

```
A>PASSWORD c:\handsoff\*.bas /W:simon /S
```

PATH

Format

```
PATH [/H] [[d:]dirpath [;[d:]dirpath]... | ;]
```

Explanation

PATH sets a search path for commands or batch files not found by a search of the current directory. If a command cannot be found in the current directory, the operating system searches the named directories in the order you entered them, but does not change the current directory. The maximum length of a search path is 123 characters.

PATH with no parameters displays the names indicated in the previous PATH command (the search paths currently defined).

If you type PATH with a semi-colon, PATH ;, you reset the search path to *no extended search path* or a “null” path. If no path is specified, the operating system only searches in the current directory of the current drive.

Incorrect information in the paths will not be found until the specified paths are searched. If a specified path does not exist, the operating system ignores that path and goes to the next.

If the command entered is not found in any of the directories specified in PATH, the message:

```
Command or filename not recognized
```

is displayed.

Example

In the following example, **spell.com** is contained in directory **mydir** on drive B:, and the default drive is drive A:

```
A>PATH a:\sub1;a:\sub1\sub2;b:\mydir
```

This command sets the search path so that when you type SPELL at the command line the operating system looks into the current directory of the current drive specified, followed by **a:\sub1**, then **a:\sub1\sub2**, then **b:\mydir**, until **spell.com** is found and loaded.

PCTERM

Format

PCTERM [/H]

Explanation

PCTERM is a terminal emulator that runs under a DOS operating system on a PC and allows the PC to act as a terminal on a multiuser computer. The PC user can then run local DOS programs when the PC is not required to act as a terminal.

To run the emulator, the program **pcterm.com** must be copied to a disk on the PC from the operating system distribution diskette. To start the emulator, make that drive the active drive, then type:

PCTERM

and press the **Enter** key.

A menu is displayed with three options:

- ☐ **F1**, to start the emulator.
- ☐ **F2**, to set terminal parameters.
- ☐ **Esc**, to leave PCTERM and return to the DOS prompt on the PC.

For more information about PCTERM, see Chapter 3, “Adding a PC as a terminal.”

One of the parameters can be set so that PCTERM bypasses the menu and goes directly in to emulator mode when you start PCTERM.

You can return to the menu at any time by pressing the **Ctrl+Alt+F10** keys.

PIFED

Format

```
PIFED [/H] filespec [/B] [/Ilabel=setting] [/R]
```

Explanation

PIFED allows you to create or modify the program information embedded in a DOS **.exe**, **.com** or **.cmd** file. This program information allows the operating system, for example, to allocate the correct amount of memory for the program and to prevent corruption of interrupt vectors.

There are two ways to run the PIFED command, using the PIFED screen or by typing all the settings at the command line.

Command switches

/B	Reverts to black and white display in menus. This may improve the clarity of the display on some computers.
/Ilabel=setting	Installs program information in the specified file.
/R	Remove program information from the specified file.

For more information on the PIFED command see Chapter 15, “Customizing your system.”

PRINT

Format

```
PRINT [/H] [wildspec] [/B:buffsize] [/C] [/D:device]
[/M:maxticks] [/P] [/Q:queuesize] [/S:timeslice] [/T]
[/U:busyticks]
```

Explanation

PRINT is used to print text files.

When you have created files on disk that need printing, you can use PRINT to manage the print process. PRINT creates a list of files called a *print queue* and then prints the files one by one.

PRINT runs in the background, leaving your computer free so you can carry on working as normal. PRINT is an external command, but part of it remains loaded in memory. When you start the PRINT command, you see the message:

```
Resident part of PRINT installed
```

Command switches

<i>/B:buffsize</i>	Sets the size of the print buffer (in bytes). The default is 512. A larger buffer can (sometimes) improve the print speed. The buffer size can only be set the first time the PRINT command is used during a working session.
<i>/C</i>	Deletes the named file(s) from the print queue. Filenames can be typed before or after this option in the command line: the option looks at the one filename immediately preceding it in the command line and at all the filenames typed after it.
<i>/D:device</i>	Sends the printer output to the named device. The default is PRN (which is the same as LPT1). The other valid devices you can name are: AUX, COM1, COM2, LPT1, LPT2, and LPT3. The device name can only be specified the first time the PRINT command is used during a working session.

Command switches (continued)

/M:maxticks These parameters work together to govern what portion of computer time PRINT is allowed to use. *Timeslice* (default 8) describes how often the computer gives control to the PRINT utility; *maxticks* (default 2, range 1 through 255) governs for how many clock-ticks PRINT runs once it has control; *busyticks* (default 1) specifies how long PRINT waits before handing back control if the printer is busy. (A clock-tick is approximately 1/18th of a second.)

Note

Each of these parameters can only be specified the first time you issue the PRINT command during a working session.

/P Adds the named files to the print queue. The rules for using it are the same as those governing the */C* option.

/Q:queuesize Sets the maximum number of files that can be held in the print queue. The default is 10, in the range 4 to 32. This parameter can only be defined the first time you use PRINT during a working session.

/T Deletes all the files from the print queue. The printer prints the message:

All files cancelled by operator

Note

See Chapter 13, "Printing," for information on the print spooler, CSPOOL.

Examples

If you type this command when using PRINT for the first time after starting up the system:

```
C>PRINT /Q:20 chap1.doc
```

the command sets the spool queue size to 20. As the /D option was not given, you get this message:

```
List device? [PRN]
```

You need to press **Enter** to print via the default port LPT1 (LPT1 is equivalent to PRN). Type the port name if you want to print via another port. If PRINT was already started, using the /Q option would result in an error message.

The following command deletes the files **chap2.doc** and **chap3.doc** from the print queue:

```
C>PRINT chap2.doc /C chap3.doc
```

PRINT allows you to combine several commands on one command line.

The following command deletes all the files on the print queue and then adds files **newbook1.doc** and **newbook2.doc** to the queue:

```
C>PRINT /T newbook1.doc newbook2.doc
```



PRINTER

Format

PRINTER [/H] [n]

Explanation

The **PRINTER** command displays or sets the PRN device assigned to your current DOS session. Each DOS session can have a different printer assignment or all DOS sessions can share access to a single printer.

The **PRINTER** command is supported, but has been superseded by the **PRINTMAP** command.

PRINTMAP

Format

```
PRINTMAP [/H] [LPTn=PRN[n] | NUL] ... [COMn=AUX[n] | NUL] ...  
[/A] [/P] [/*]
```

Explanation

The operating system assigns all printers numeric values (shown as PRN*n*), and assigns all serial ports numeric values shown as AUX*n*. However, most DOS applications access printers by using device names LPT1, LPT2, LPT3, and access serial ports by using device names COM1 and COM2.

The PRINTMAP command allows you to establish which printer or serial port is assigned to which DOS device name and allows you to display the current assignments. If you want these assignments to apply permanently, put the PRINTMAP assignment(s) in your **autoexec.bat** file.

If you type PRINTMAP on its own, you see a display of the current mappings for the current DOS session. The display also has a brief description of each of the devices. The descriptions are taken from a text file called **printmap.dat**, in the same directory as the **printmap.exe** file. A file containing default descriptions for each possible printer and auxiliary device is supplied with the system. You can edit this file using an ASCII editor such as EDITOR, so that the descriptions match your own devices.

Note

PRINTMAP assignments are only valid for the current working session. Place assignments in **autoexec.bat** if you want them to apply permanently.

Changing printer assignments

To change the assignment of your printers use the format:

```
PRINTMAP LPTn=PRN[n]
```

The printers are identified as PRN, LPT1, LPT2 and LPT3. PRN always maps to LPT1.

Map to the NUL device to discard output from the specified printer port.

Changing serial port assignments

To change the assignment of your serial ports use the format:

```
PRINTMAP COMn=AUX[n]
```

The serial ports are numbered COM1 and COM2. AUX always maps to COM1.

Map to the NUL device to discard output from the specified port.

Listing printers and auxiliary devices

You can use the PRINTMAP switches to find out what printers and serial devices are available.

Command switches

/A	Displays a list of available serial devices.
/P	Displays a list of available printers.
/*	Allows you to change the assignment for all the DOS sessions on your terminal at once. Without the /* option only the assignments for your current DOS sessions are changed.

Examples

To display the current settings for your current DOS session, type:

```
C>PRINTMAP
```

A display similar to the following appears:

```
LPT 1  Printer 0   Parallel Port 0
LPT 2  Printer 1   Parallel Port 1
LPT 3  Printer 2   Parallel Port 2

COM1   Aux      0   COM1 Port
COM2   Aux      1   COM2 Port/Multiport 1
```

To assign PRN3 to LPT1 and AUX0 to COM2, for all DOS sessions on your terminal, type:

```
C>PRINTMAP LPT1=PRN3 COM2=AUX0 /*
```

PROMPT

Format

PROMPT [/H] [promptstring]

Explanation

The PROMPT command modifies the command prompt. You can set the command prompt to contain any valid ASCII characters or certain special character strings.

Type PROMPT with no parameters to reset the command prompt to the default system prompt.

The *promptstring* can be made up of any of the valid ASCII characters or the following special characters, preceded by a dollar sign (for example \$t):

Character	Meaning
b	The character.
d	The date.
e	The ESCape character.
g	The ">" character.
h	A backspace; the previous character is erased.
l	The "<" character.
m	A message notifying you if you have mail.
n	The default drive letter.
p	The current directory path of the default drive.
q	The = character.
t	The time.
u	The login name of the user.
v	The version number.
\$	The \$ character.
- or _	The CR LF sequence (Carriage-return and line feed, go to beginning of new line of the screen).

Any other values are considered invalid and ignored by PROMPT (for example \$c).

Examples

The following command sets the system prompt to HELLO:

C>PROMPT hello

The following command sets the system prompt to the default drive and path plus the > character:

C>PROMPT \$p\$g

The following command sets the system prompt to display the time and date:

C>PROMPT time = \$t\$_date = \$d

The system prompt appears as follows:

TIME = *current time*

DATE = *current date*

REBOOT

Format

REBOOT [/H]

Explanation

The REBOOT command is a synonym for **Ctrl+Alt+Del** and performs the function assigned to these keys on the main console, either performing a system reset, or stopping the current program.

If system security has been enabled, REBOOT is only available to the Superuser. If system security is not in use, you may wish to protect this command with a password. Use the PASSWORD command to do this.

Caution Use this command with care.

Using REBOOT to perform a system reset stops work on all terminals. Any running programs get stopped, and users may lose their work.

RECOVER

Format

```
RECOVER [/H] d:|wildspec
```

Explanation

The RECOVER command enables you to retrieve information from files that have been corrupted. If you have a file that is only partially readable, RECOVER removes the unreadable parts so you can access the rest. It has two forms: you can either recover individual files or groups of files, or you can recover an entire disk.

WARNING

Only recover a disk if your *root directory* is corrupted, because if you only specify a disk, RECOVER assumes the *entire* disk is scrambled and replaces your root directory with a *new* one. All the information contained in your files is reorganized into a series of files as described above, and you then have to sort out where your data is and restructure it.

Recovering a disk

To recover an entire disk you type:

```
RECOVER d:
```

Each recovered file is converted into a file in the root directory. These files have filenames in the form *filexxxx.rec*, where *xxxx* specifies a number starting at 0000.

Note

You may need to repeat the operation if there are more cluster chains to convert than there are available entries in the root directory. In this case you should copy the first set of recovered files to another directory on another disk, delete them from the root directory, and then re-issue the command.

Recovering individual files

You can use RECOVER to recover single files or a specific group of files (you can use wildcards in the specification). For example:

```
C>RECOVER c:\reports\invoice
```

In this case each file in the path that matches the specified filename, for example **invoice**, is checked for bad areas (or “blocks”). If the operating system finds any bad blocks they are skipped over and marked as bad blocks in the File Allocation Table. The record of the file’s length (shown when you use the command DIR) is also modified to reflect the new length of the file.

Once you have recovered the file or files, you can edit them and re-enter lost information.

Remarks

If you specify a subdirectory in the command syntax, it means that the files in that subdirectory are recovered and copied to the *root* directory, not to the original subdirectory where the files were stored.

RECOVER is most useful when you are attempting to recover *text* files.

RECOVER does not work on networked drives. Nor does it work on drives affected by a JOIN or SUBST command.

Example

This checks whether there are any bad blocks in the files matching **salesm?.doc** in directory **reports** on drive C:. Any bad blocks are marked in the File Allocation Table and the record of the files’ length is modified accordingly.

```
C>RECOVER c:\reports\salesm?.doc
```



RENAME (REN)

Format

RENAME [/H] oldwildspec newwildspec

Explanation

The RENAME command lets you change the name of a file or move a file between subdirectories on the same disk. RENAME can be shortened to REN.

Renaming a file

If the file to be renamed is not on the current drive or in the current directory, you must precede the old file specification with the correct drive/path specification. The following command line renames **myfile.typ** as **file1.txt**:

A>REN b:myfile.typ file1.txt

On password-protected files, you must enter the default password before renaming a file. The same file password is then required to access the new file specification. You can remove the password protection on the file before renaming it (see the PASSWORD command).

If system security is in use, you can only rename files to which you have delete access.

Wildcards can be used, for example the command:

A>REN *.in *.out

renames all files with the file extension **.in**, so that they have the file extension **.out**.

RENAME does not let you rename a file to an existing name. If you attempt to do this the message:

File already exists

is displayed, and the command prompt returned. Any files designated to be renamed remain unchanged.

Moving files

RENAME can be used to move files between subdirectories on the same disk. Use it instead of the COPY and DEL commands.

For example the command:

A>REN myfile.lst \wordproc\memos

moves **myfile.lst** to the directory **a:\wordproc\memos**; **myfile.lst** no longer exists on the original directory.

REPLACE

Format

```
REPLACE [/H] [@]wildspec dirpath [/A] [/H] [/M] [/N]  
[/P] [/R] [/S] [/U] [/W]
```

Explanation

REPLACE copies selected files from a source to a destination. It is useful for updating previous versions of files. For example, you can *add* files to a destination directory or you can replace only files that have been changed since the last copy. Use the /A switch to add files to a subdirectory.

The *wildspec* is the drive, path and name of the source files; *dirpath* shows the drive and path where you want the files to be placed.

As well as replacing single files in one directory, you can also replace files across subdirectories, using the /S switch.

Command switches

/A	Copies only files not already on the destination.
/H	Does not ignore files with hidden or system attributes.
/M	Merges changed files on the source with unchanged files at the destination.
/N	Preview operation; does not actually copy any files.
/P	Prompts before copying each file.
/R	Overwrites read-only files on the destination with fresh copies from the source.
/S	Looks in subdirectories of the destination for files that match the <i>wildspec</i> .
/U	Only copy files that are older on the destination than on the source.

Command switches (continued)

/W Waits for disks to be changed before starting to look for the source files, and waits before starting to copy them to the destination. This allows the REPLACE program to be loaded from a diskette, the diskette changed for the source diskette, and then changed for the destination disk.

Example

The REPLACE option is especially useful for updating software. For example, you might have a program called **wcounter.bas** in various subdirectories on your hard disk. If you change the copy of **wcounter.bas** in the **\wordproc** directory, you can copy the changed version of the file to all the other places in the hard disk by using the following command:

```
C>REPLACE a:\wordproc\wcounter.bas c:\ /S
```

RESTORE

Format

```
RESTORE [/H] source_drive destination_drive
[filespec] [/A:date] [/B:date] [/E:hh:mm:ss]
[/L:hh:mm:ss] [/M] [/N] [/P] [/R] [/S]
```

Explanation

Restores directories and files that were backed up with the BACKUP utility. *Source_drive* is the drive containing the backed up files, and *destination_drive* is where you want to restore your files to. RESTORE prompts you for each backup diskette in turn until all the files you specified are restored.

If system security is enabled, and the owner of each file was recorded by using the /U option in the BACKUP command, the files are restored to their original owner(s).

Command switches

/A:date	Restores all the files that have been altered on the original disk on or after the date specified. The format of the date depends on the current country code (see the DATE command).
/B:date	Restores all the files that have been altered on the original disk on or before the date specified. The format of the date depends on the current country code (see the DATE command).
/E:hh:mm:ss	Restores all the files that have been altered on the original disk at or earlier than the time specified.
/L:hh:mm:ss	Restores all the files that have been altered on the original disk at or later than the time specified.
/M	Restores all the files that have been altered on the original disk, or deleted, since the backup was taken.
/N	Restores files that do not exist on the destination.

Command switches (continued)	
/P	Prompts you for permission to restore read-only files, or files that have changed since the last backup.
/R	Reports which files would be restored, but does not take any action. This is useful for checking that you will restore the files that you want. This option also checks the integrity of backup disks.
/S	Restores all subdirectories associated with the current directory on the destination disk. RESTORE will create the required subdirectories on the destination disk if necessary.

Remarks

RESTORE reports any errors during the restoration process using the codes shown in the table that follows.

Table 11-11 RESTORE Error Codes

Code	Explanation
0	Normal completion.
1	No files found to restore.
2	Some files not restored due to file sharing conflict.
3	Restoration process ended by a Ctrl+Break .
4	Restoration process ended by an error.

This feature can be useful in creating batch files with an IF ERRORLEVEL statement in them to improve the process (see Chapter 10, “Batch processing.”)

Do not use the RESTORE program with disks affected by an ASSIGN or SUBST command.

Examples

This example restores a backup of all files on all directories on drive C:.

```
C>RESTORE a: c:\ /S
```

This command restores the file **letter.doc** in the subdirectory **\backedup** on drive C:

```
C>RESTORE a: c:\backedup\letter.doc
```

RMDIR (RD)

Format

RMDIR [/H] [d:]dirpath

Explanation

The RMDIR command removes a specified subdirectory. The subdirectory to be deleted must be empty except for the . and .. markers which refer to the directory itself, and its parent directory; these entries cannot be deleted. RMDIR can be abbreviated to RD.

Before you remove a subdirectory, ensure that:

- ☐ The subdirectory contains no files or additional subdirectories.
- ☐ The subdirectory is not the current directory on any drive in any DOS session.
- ☐ The subdirectory is not being used by a SUBST command.

Remarks

If you get the error message:

Directory not empty or in use

but when you check the relevant subdirectory only the . and .. entries can be seen, check again with the XDIR command. The XDIR command will show if any hidden, system, or password-protected files are present.

Examples

This command erases subdirectory **france** on drive A:.

A>RMDIR clients\foreign\france

This command erases subdirectory **accts.rcv** on drive C:.

A>RD c:\finance\accts.rcv

SECURITY

Format

SECURITY [/H] [/B]

Explanation

SECURITY lets the Superuser on a secure system set up new users, display and change details of existing users, and delete users. It also lets users restrict access to a hard disk partition. SECURITY is a menu-driven utility. Select the option you require, then follow the instructions shown on the screen. Press **F1** for contextual help as you proceed through the utility. User account information is stored in two files, **user.sys** and **user.inf**.

Command switch

/B	Recognizes the monitor as black-and-white, and presents menus accordingly.
----	--

The table that follows summarizes the information that SECURITY requires in order to establish a new user account.

Table 11-12 User Account Information

Field	Meaning
Group name	Each user must belong to a group, and the group is identified first.
Group id	Each group is given a unique identifying number (#00 to #FF). The next available number is displayed.
Login name	Each user must be given a login user name.

**Table 11-12 Information Required for New Users
(continued)**

Field	Meaning
User id	Each user is given an unique identifying number (#00 to #FF) in the user's group. The next available number is displayed.
Login password	Enter the password which the user must enter to log in for the first time. (The user should then change the password with the PASSWORD command, so that no-one else knows it.)
Default access permissions	The access permissions that will apply to the files the user creates (until they are changed by the user with the XATTRIB command).
Home drive/directory	These two fields define the current directory when the user has logged in. What you enter depends on how you organise the system, perhaps one directory per user or one per group with a subdirectory for each user.
Comments	An optional field in which you can enter a brief comment.

Note

User names, groups and passwords can be up to 12 characters long. User and group names must be unique - there cannot be a user **accounts** and a group **accounts**, nor can there be an **accounts** in two groups. User names, groups and passwords are case insensitive, so **ACCOUNTS** is taken to be the same as **Accounts**. However, the name is recorded exactly as you enter it and always displayed that way.

SET

Format

SET [/H] [name=[parameter]]

Explanation

SET inserts variable strings into the command processor's environment. All the strings in the environment are available to all commands and applications.

The string (beginning with *name*) is inserted into a part of memory reserved for environment strings. Lowercase letters in the name are converted to uppercase when added to the environment. If the name was in the environment, it is replaced by the new parameter.

If you type SET with no options, you see a list of the strings already in the environment.

To delete a name from the environment, type SET with a variable name and no parameter.

Remarks

Among the environment variables set automatically when the operating system is started are:

COMSPEC

OS

VER

In "secure" systems with user accounts established the following variables are also present:

USERNAME

GROUPNAME

COMSPEC defines the path to the command processor, **command.com**. It is used by programs that sometimes look for the command processor; typically programs that allow you to access operating system commands without leaving the program.

Command.com is provided with the operating system; you can, however, insert your own command processor instead by setting a new COMSPEC.

OS describes the operating system *type*. This environment string is useful for distinguishing between operating systems. For example, if you run two or more different operating systems at different times on one computer, you can devise a batch file that looks for the OS environment string, finds which operating system is present, and then takes appropriate action. SET can be used in your **autoexec.bat** file to automatically set strings when you start the operating system.

Examples

To set a string including the console number, you might include the following statement in your **autoexec.bat** file.

```
SET CONSOLE = %1
```

Other examples of SET that could be found in **autoexec.bat** are:

```
SET GLOBAL_CONSOLE = %1
```

```
SET CSPOOL = c:\osutils\spool
```

SETUP

Format

SETUP [/H]

Explanation

SETUP is a menu-driven program that allows you to change system characteristics and save them so that they become the default values when the system is next restarted.

To start the SETUP program, type:

SETUP

and press **Enter**.

Context sensitive online help is available for each screen and option. For more information about SETUP, see Chapter 16, "Customizing your system."

SHARE

Format

SHARE [/H]

Explanation

SHARE is a dummy utility and is provided for compatibility with other DOS operating systems. It is required during the installation of some DOS applications.

The operating system automatically provides support for file locking, (a means by which a file, or part of a file, may be reserved for unique access by one program) and so allows file sharing by default.

SHOW

Format

SHOW [/H] [/A] [/D] [/L] [/M] [/P] [/U] [/W]

Explanation

Use SHOW to display disk drive size, available space and other system parameters. The table that follows lists the SHOW options.

Table 11-13 SHOW Options

Option	Description
/A	Displays all the information covered by the other options.
/D	Displays free space on and the size of all logged in drives.
/L	Displays printer information.
/M	Displays information about how memory is configured.
/P	Displays the information one screenful at a time.
/U	Displays the processes currently running on the system.
/W	Shows who is logged in (on systems with security enabled).

SID

Format

SID [/H]

Explanation

SID is a symbolic debugger program. The use of the debugger and the SID commands are described in Chapter 14, “The SID debugger.”

SORT

Format

```
SORT [/H] [/R] [/+n]
```

Explanation

SORT is a filter program which reads standard input, sorts it alphabetically, and then writes it to the standard output device.

Use the redirection symbol < or the pipe symbol | to filter data from a filename or command through the SORT program. Use the > redirection symbol to send output to a file or printer.

Note

The SORT command does not differentiate between upper- and lowercase letters.

Command switches

/R	Sorts in reverse order, that is 9 to 0, or Z to A.
/+n	Makes the sort start on the “nth” column of a line. If omitted, the sort starts from the default, column 1.

Remarks

For more information on redirection, piping and the SORT command, see Chapter 12, “Redirecting input and output.”

Examples

The following example reads and sorts the file **index.dis** and writes the output to the printer:

```
A>SORT<index.dis>PRN:
```

The following example uses SORT to make up a directory listing in *reverse* alphabetical order, and then stores it in the file **reverse.ord**.

```
A>DIR | SORT /R >reverse.ord
```

STOP

Format

STOP [/H] [programname sessionnum]

Explanation

The STOP command is used to terminate a running program when it cannot be halted in the normal manner.

STOP is an internal program and can be run from any DOS session. It is used to stop any program that is running on another DOS session on that same terminal. Notice that this form requires the entry of the program name and the DOS session number in which it is running.

Note

The simplest way to stop a single DOS session on a terminal is by entering **Ctrl+Alt+Del**. On the main console use **Ctrl+Alt+Del** with caution, because it may have been set up to reset the complete system.

SUBST

Format

```
SUBST [/H] [d: d:dirpath | [d:] /D]
```

Explanation

SUBST allows you to make a subdirectory appear as if it were another disk drive. You do this by replacing a drive-and-path specification with a single drive letter. So, for example, you can replace a path like **\wordproc\simon\letters** with the drive F:. You can then use drive F: in your commands as if it was a real disk drive. This saves you from having to type in long path specifications over and again, and it can be used for programs that do not recognize subdirectories.

Typing SUBST without any parameters displays the currently substituted drives.

Use the /D parameter to remove substitutions when you have finished working with them. /D operates on the specified drive, or on all substitute drives if none is specified on the command line.

Caution When using substituted drives you should avoid using commands that involve entire disks, such as:

BACKUP	FORMAT
CHKDSK	LABEL
DISKCOMP	RECOVER
DISKCOPY	RESTORE
FDISK	SYS

Always switch off a substitution when you have finished working with it.

Examples

The following example assigns drive F: to the path **\wordproc\simon\letters** on drive C:.

```
A>SUBST f: c:\wordproc\simon\letters
```

The example below removes the assignment from drive H:.

```
A>SUBST h: /D
```

SUSPEND

Format

SUSPEND [/H] [[=]ON|OFF]

Explanation

Use the `SUSPEND ON` command to temporarily stop a program with the **.exe** or **.com** file extension while it is in a background DOS session. Program operation resumes when the program returns to the foreground DOS session.

(Use `PIFED` to suspend the operation of programs with the **.cmd** filetype.)

The only reason to suspend a program running in background mode is that it may be taking up processor time.

`SUSPEND` works on a single DOS session basis. Therefore, you must run `SUSPEND` in the DOS session you are going to use for that program. The `SUSPEND ON` state, once initiated, remains in effect on each DOS session until the computer is turned off, reset, or you enter `SUSPEND OFF`.

Remarks

The `PIFED` command can be used to suspend programs automatically when the DOS session they are running in is not displayed. This overrides the current `SUSPEND` setting for the DOS session.

SYS

Format

SYS [/H] d:

Explanation

Transfers the two main operating system files and **command.com**, from the default drive (that is, the drive on which SYS is being run) to a specified destination drive (d:).

SYS allows you to transfer the operating system files to a diskette to make a drive bootable (some program diskettes need the operating system on them to work), or to change them to match the ones on the default (or SYS) drive.

SYS also transfers the command processor defined by COMSPEC to the root directory of the destination drive.

Note

If you wish to set a hard disk partition to "secure," by means of the SECURITY utility, you must have the system files present in the partition.

Remarks

The operating system does not require space to be reserved for its system files on a disk, or that they be placed at the beginning of the disk; therefore you should have no problems making a disk bootable.

SYS does not work on drives affected by a SUBST command or across networks.

Examples

To transfer the system files from drive C: to drive A:, type:

C>SYS a:

To transfer the system files from drive A: to drive C: (you must have booted from drive A:), type:

A>SYS c:

TIME

Format

```
TIME [/H] [hh[:mm[:ss]]] [/C]
```

Explanation

The TIME command displays and sets the time.

Your computer has a real-time battery-backed clock so it should not be necessary to set the TIME or DATE whenever you start the system.

Normally, you will only use the TIME command to display the time of day. (The operating system also uses time to record the time a file is created or updated.)

Command switch

/C	Displays the time continuously, until you press any key to exit.
----	--

The clock works on a 24-hour basis. This means that 1 p.m. is 13:00 and midnight is 00:00.

To check the current time setting, invoke the command without parameters:

```
A>TIME
```

TIME displays the time and prompts you as follows.

```
Time: hh:mm:ss.xx
Enter time:
```

Press **Enter** to keep the current values.

If you need to enter a new time, enter the hours, minutes and seconds and then press **Enter**. Note that the last value shown above, *xx*, indicates hundredths of a second. You can also enter a new value for time on the command line.

Time values are defined as follows:

- ❑ *hh* is a one- or two-digit number from 0 to 23 indicating the hour.
- ❑ *mm* is a one- or two-digit number from 0 to 59 indicating the minute.
- ❑ *ss* is a one- or two-digit number from 0 to 59 indicating the second.

TIME assumes that the first entry specifies the hour. The minutes and seconds are optional, and if they are not entered are set to zero. You must use a colon to separate the entries.

If you enter a number that is out of range, TIME returns the message and prompt:

```
Invalid time specified
Enter time:
```

Examples

The following command sets the time for 30 minutes and 59 seconds after 10 pm:

```
A>TIME 22:30:59
```

The following command sets the time at 12 minutes after midnight:

```
A>TIME 00:12
```



TOUCH

Format

```
TOUCH [/H] [@]wildspec... [/D:date] [/F:E|J|U] [/P]  
[/R] [/S] [/T:hh:mm:ss] [/U:name]
```

Explanation

TOUCH resets the time and date stamps for a single file or a group of files. If you change the date stamps on a group of files you can also alter the date format using the /F switch.

Command switches

<i>/D:date</i>	The date to be set. The default is the current date. The date format depends on the current country code, set up in the cconfig.sys file or the value specified in the /F: option.
<i>/F:E J U</i>	Specifies European, Japanese or USA date format for the date option before touching each file. This overrides the default format determined by the current country code.
<i>/P</i>	Prompts before touching each file.
<i>/R</i>	Includes read-only files in the touch operation.
<i>/S</i>	Touches files in subdirectories associated with the current directory.
<i>/T:hh:mm:ss</i>	The time to be set. The time is set in 24 hour clock with midnight being 00:00. The default is the current time.
<i>/U:name</i>	Limits the effect of the command to files belonging to the user or group named.

Examples

The following command resets the date stamps for all the files **states*.*** using the US date format, overriding the date format set up in the **cconfig.sys** file:

```
A>TOUCH states*.* /D:06-22-89 /F:U
```

TREE

Format

```
TREE [/H] [d: | wildspec] [/B] [/F] [/G] [/P]  
[/U:name]
```

Explanation

The **TREE** command displays the hierarchical path of directories and subdirectories on a disk, or if used with a *wildspec* finds all the files matching the *wildspec* on the disk.

Command switches

/B	Displays in brief mode which leaves out the number of files in each directory.
/F	Displays all the files in the subdirectories, as well as the subdirectories themselves.
/G	Displays directory structure in graphical format, that is lines are drawn to show the directory structure.
/P	Pauses after each screenful in a long TREE display.
/U:name	Displays only the files belonging to the user <i>name</i> or to the group <i>name</i> .

TYPE

Format

TYPE [/H] wildspec [/P]

Explanation

The **TYPE** command displays the contents of a text file on your screen (if you try to display a program or binary file, you just see unintelligible characters).

Command switches

/P Causes the text to be displayed a screenful at a time; move to the next screen by pressing any key. Press **Ctrl+C** to abort the display. Alternatively you can press **Ctrl+S** to stop the display from scrolling. To restart scrolling, press **Ctrl+Q**.

Press **Ctrl+P** to have the file contents sent to the printer as well as the screen. Press **Ctrl+P** again to stop output to the printer.

If the file is password-protected for **READ** mode, you must set the global password, or add the password to the end of the *wildspec*. Wildcard characters can be used to display multiple files. When you type the name of the file you must enter the full name and extension.

Examples

This command displays the text file **memo.txt** on the screen a screenful at a time.

C> TYPE memo.txt /P

This command displays the contents of all files with the extension **.bat**.

C> TYPE *.bat

Password-protected files

VER

Format

VER [/H]

Explanation

VER displays the version of the operating system that you are using.

VERIFY

Format

VERIFY [/H] [=]ON|OFF

Explanation

VERIFY ON causes the operating system to check that data is written to the disk correctly after each write operation. This, however, causes your system to run slightly slower and so the default value is VERIFY OFF.

Entering VERIFY with no parameters shows whether VERIFY is currently ON or OFF.

If you type VERIFY ON, the verify setting stays on until you or a program specifically switches it off.

Examples

To check the current status of VERIFY, type:

C>VERIFY

To switch VERIFY off, type:

C>VERIFY OFF

VOL

Format

VOL [/H] [d:]

Explanation

VOL displays the volume label of the specified drive, or of the default drive if no drive is given.

If the disk in the drive has no label then the message:

Volume in drive *d* does not have a label
is shown.

Example

The following command displays the volume label on the current drive C::

C>VOL

Volume in drive C is Accounts

XATTRIB

Format

```
XATTRIB [/H] [@]wildspec [/O|G|W:R|W|D] [/A:name]
[/B] [/D] [/F] [/P] [/S] [/U:name]
```

Explanation

XATTRIB is only used when system security is enabled. It is used to display and set the access permissions of existing files and directories, to set default access permissions and to change the ownership of a file or directory.

Notes

Users can only change the access permissions and surrender ownership of their own files and directories. However, the Superuser can perform these functions on any file or directory in the system, including those belonging to other users.

You cannot give greater access permissions to the "world" than to the group or owner. Similarly, you cannot give others in your group greater access permissions than you have yourself.

Command switches

/O G W	Defines access permissions for: /O, the owner /G, the owner's group /W, the "world" (everyone else).
:R W D	Defines access permissions in terms of: R, read access W, write access D, delete access.
/A:name	Changes the file owner to be the user with the login name <i>name</i> .
/B	Displays filename only.
/D	Operates on directories not on files.
/F	Changes the default access permissions for newly-created files.

Command switches (continued)

/P	Prompts the user before operating on each file.
/S	Operates on all the files in the subdirectories of the current directory as well as the current directory itself.
/U:name	Operates only on the files belonging to user <i>name</i> or to the group <i>name</i> .

Examples

To display the current access permissions of all files in the current directory, type:

```
C:\USER\DOC>XATTRIB
```

```

rwd --- --- fred      accounts      C:news.doc
rwd rwd --- joanne    accounts      C:joanne.txt
rwd --- --- fred      accounts      C:personal.doc
rwd rw- r-- fred      accounts      C:letter.doc
```

The access permissions, owner, group and file name are given for each file. Note that files belonging to other users are displayed, even if you do not have any access permissions to them. (The full file name including the path is only shown if the **/S** switch is used.)

The following command sets the access permission of the file **news.doc** so that any user can read the file, but only users in the owner's group can write to the file. Note that this command could only be executed by the owner (**fred**) or by the Superuser:

```
C:\USER\DOC>XATTRIB news.doc /G:RW /W:R
rwd rw- r-- fred      accounts      c:news.doc
```

The following command gives the file **letter.doc** to the user with the login user name SUSAN. Note that the file access permissions remain the same, and that the file's location does not change. This command could only be executed by **fred** or by the Superuser:

```
C:\USER\DOC>XATTRIB letter.doc /A:susan
rwd rw- r-- susan      admin          c:letter.doc
```

The following command displays the names of all the files in the current directory belonging to the accounts group in the brief format:

```
C:\USER\DOC>XATTRIB /U:accounts /B  
C:news.doc  
C:joanne.txt  
C:personal.doc
```

The following command sets the default access permissions for users in your group to read and write access (note that owner and world permissions are not changed):

```
C:\USER\DOC>XATTRIB /G:RW /F  
Default file create permissions for Sue: rwd rw- r--
```

XCOPY

Format

```
XCOPY [/H] [@]wildspec [dirpath] [/A] [/D:date] [/E]  
[/H] [/M] [/P|/C] [/R] [/S] [/U:name] [/V] [/W]
```

Explanation

XCOPY (or Extended Copy) selectively copies groups of files. It can be used to copy whole subdirectories.

The *wildspec* is the drive, path and name of the files to be copied.

Dirpath is the destination drive and path to which the files are to be copied.

Command switches

/A	Copies only those files that have the archive attribute set. The attribute of the source file is not changed. (See the ATTRIB command explanation in this chapter, for an explanation of the archive attribute and other attributes.)
/D:date	Copies only those files whose date is the same or later than the date specified. The format of the date depends on the current country code (see the DATE command).
/E	Creates subdirectories at the destination even if they end up being empty after all the copying is complete.
/H	Includes hidden and system files. (Files with the system attribute are explained in the ATTRIB command description.) The default is for system files to be ignored.
/M	Copies files that have the archive attribute set and resets the attribute bit of the source file.
/P or /C	Prompts you to confirm each file before it is copied.

Command switches (continued)

/R	Overwrites read-only files.
/S	Copies the files in the source directory and files in all the subdirectories below it.
/U: <i>name</i>	Copies only those files that belong to the user or group <i>name</i> . This option can only be used if system security is enabled.
/V	Verifies that the data written to the destination is a true copy of data in the source.
/W	Waits for a diskette to be inserted before beginning to search for source files and (if, necessary) wait again before the files are written to the destination. Press a key to continue or Ctrl+C to stop the copying. This allows the XCOPY program to be loaded from diskette, the diskette changed for the source diskette, and then changed for the destination diskette.

Remarks

You can use XCOPY to rename files while copying them in the same way as the COPY command, by specifying a new file name for the destination file.

Using XCOPY you can duplicate a tree of subdirectories and files on another disk. If you want an *exact* copy of a tree structure, you need to specify the /E switch as well as the /S switch to ensure all subdirectories including empty ones are copied.

You can use XCOPY to copy a subdirectory from one type of disk to another disk with a different format.

Password-protected files

XCOPY can be used with password-protected files and directories; you are prompted for the password. For example, if the subdirectory **sales** was assigned the password **magic**, and you entered the following command from the root directory:

```
C>XCOPY c:\sales a:
```

you are prompted:

```
c:\sales ...path password?
```

Type in the password and press **Enter** (it is not shown on the screen), and the contents of the subdirectory **sales** are copied to the diskette in drive A:.

Examples

Suppose you have files in a directory called **c:\wordproc** which contains two subdirectories **\wordproc\jenny** and **\wordproc\olivia**. You want to copy all the files that have the file extension **.let** from these directories on drive C: onto a diskette in drive A:. You also want the appropriate subdirectories to be created on the diskette in drive A:. You type:

```
C>XCOPY \wordproc\*.let a: /E /S
```

The next example copies all the files that are marked for archiving from a diskette in drive A: to another diskette in drive B:. (Note that the archive attribute is set by the operating system whenever a file is written to and then closed.) You are prompted to insert a diskette in drive A: before the copying operation starts. The archive attribute is turned off at the source file to show that the file has been copied. You type:

```
A>XCOPY *.* b: /M /W
```

The following command copies all the files listed in the file **file.lst**, stored in the directory **c:\fred**, to the directory **d:\john**. Each file specification in **file.lst** must start on a new line, or be separated by a space, tab or comma:

```
C>XCOPY @c:\fred\file.lst d:\john
```

XDEL

Format

```
XDEL [/H] [@]wildspec... [/D] [/N] [/O] [/P] [/R]
[/S] [/U:name]
```

Explanation

XDEL is a command for deleting multiple files in subdirectories and removing empty directories.

If system security has been enabled, you can only delete files to which you have delete access permissions.

If no default password is given, you are prompted for the correct one before a password-protected file is deleted.

Command switches

/D	Removes empty subdirectories.
/N	Deletes all the specified files without asking further permission. This switch is intended to be used only in batch files. If you use it in the command line, take great care that you do not delete files the wrong files.
/O	Overwrite files before deleting them. Normally, when you delete a file, the operating system amends the directory entry for that file to show the space it previously occupied is now free, but keeps the actual file contents until another file requires its space. It is then overwritten by the new file. If you want to make sure the file is really erased off the face of the disk, you can specify the /O switch.

WARNING Use the /O option with great care as once a file is overwritten, it can never be recovered.

/P	Prompts you before deleting each file.
----	--

Command switches (continued)

/R	Delete read-only files.
/S	Deletes files in subdirectories derived from the specified directory.
/U:<i>name</i>	Deletes only those files that belong to the user or group <i>name</i> . This option can only be used if system security is enabled.

Examples

The following command deletes all files with the **.doc** extension from the current directory and all the subdirectories below the current directory:

```
A> XDEL *.doc /S
path: a:\
file: *.doc
warning: FILES IN SUB-DIRECTORIES MAY BE DELETED
Is this what you wish to do (Y/N) ? Y
a:\first.doc
a:\sub1\mem1.doc
a:\sub2\mem2.doc
a:\sub3\letter3.doc
A>
```

The following command deletes all files with the **.doc** extension from the current directory that belong to the user ISMITH:

```
A> XDEL *.doc /U:ismith
path: a:\
file: *.doc
owner: ismith
Is this what you wish to do (Y/N) ? Y
a:\first.doc
a:\sub1\mem1.doc
a:\sub2\mem2.doc
a:\sub3\letter3.doc
A>
```

XDIR

Format

```
XDIR [/H] [+|-A] [+|-D] [+|-H] [+|-R] [+|-S] [wildspec]
[/B] [/C] [/L] [/N] [/P] [/R] [/S] [/T] [/U:name]
[/W] [/Z]
```

Explanation

XDIR is an expanded version of the DIR command, offering extra options for displaying the contents of directories and subdirectories.

Command switches

+ -A,D,H,R,S	Includes (+) or excludes (-) files with a certain attribute. The default is for all the files to be displayed, including system files. See the ATTRIB command for details of attributes A, H, R and S. The D attribute indicates a directory.
/B	Produces a brief directory display showing only the drive and filenames and, if the /S switch is also used, paths.
/C	Computes and displays a checksum for each file. A hash code is a four digit hexadecimal number calculated from the bytes within the file. It can be used to verify that two files are the same.
/L	Produces a long display format. This is the default.
/N	Does not sort directory alphabetically.
/P	Pauses after each screenful of display.
/R	Reorders the sort order: either the alphabetical order by filename (the default), chronological order (set by the /T option) or by file size (set by the /Z option).

Command switches (continued)

/S	Displays the files in subdirectories below the specified directory.
/T	Sorts the directory by date and time, rather than by filename.
/U:name	Displays only those files that belong to the user or group <i>name</i> . This option can only be used if system security is enabled.
/W	Displays the directory in a wide, abbreviated format. This is useful if your directory display would otherwise be too long for the screen.
/Z	Sorts the directory by file size.

Examples

The following example displays all the read-only files in the current directory and the associated subdirectories:

```
A>XDIR +r *.* /S
```

The following example displays all the files that match the specification **memo*.*** on the current directory, and sorts them into reverse chronological order, so that the most recent are at the top of the list:

```
A>XDIR memo*.* /T /R
```

The following command displays all the files with the extension **.doc** in the current directory that belong to the user ISMITH:

```
A>XDIR *.doc /U:ismith
```

XSTOP

Format

XSTOP [/H]

Explanation

XSTOP is a powerful utility that lets you:

- ☐ Display all the DOS sessions and programs currently being used on all the terminals.
- ☐ Prompt you to terminate any of the programs listed.

To terminate a program, enter the number assigned to it and press **Enter**.

WARNING Take care when using XSTOP to terminate another user's program. It is possible that you may cause the user to lose data.

To exit from XSTOP without terminating a program, press **Enter**.

You can use XSTOP to terminate any program on any DOS session throughout the system. For this reason, on a system with system security installed, we recommend that XSTOP be available only to the Superuser.

You may like to password-protect the file **xstop.exe** so that unauthorized users cannot stop other people's programs.

When you issue the XSTOP command without any parameters, you see the information listed in the table that follows.

Table 11-14 XSTOP Display Information

Field	Meaning
Entry Number	Numbers given by the XSTOP command to all programs running. The numbers are sequential: the highest number indicates the first program invoked and the lowest indicates the most recent program invoked. XSTOP always appears first on the list marked by **.

Table 11-14 XSTOP Display Information (continued)

Field	Meaning
Program Name	The name of the program running.
Console Number	The number of the DOS session in which the program is running.
Memory Size	The amount of memory used by the program.

Normally, you should use the termination command procedure described in your program's manual, because it saves the information you have entered.

Use the XSTOP command to terminate a program only if the normal termination command cannot be performed. Change DOS sessions if necessary and enter the XSTOP command. Find the program's number in the display, type that number, and press the **Enter** key.

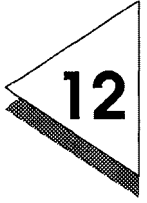
XSTOP terminates the program specified. When you return to the DOS session in which the program was running, the command line prompt is displayed and you can enter another command.

Example

This command will terminate program **myprog** running in DOS session 3:

```
A>XSTOP myprog 3
```

Redirecting input and output



NORMALLY you enter information at the keyboard, which is the standard input device. This information is sent to your screen, which is the standard output device.

It is, however, possible to redirect the flow of information, causing input to arrive from a source other than the keyboard, or sending output to a destination other than the screen. This chapter describes the various methods of achieving these ends.

12.1 Using redirection

Redirecting output

Instead of your information being sent to the screen you can send it to a file or print device. To do this you use the redirection symbol `>`.

For example, to redirect the listing that the `DIR` command produces directly to a file instead of displaying it on the screen, type:

```
C>DIR >mylist.doc
```

If the file **mylist.doc** did not exist, it would be created for you. If it does exist, the original will be overwritten, so take care.

Instead of the filename, you can enter a file specification:

```
C>DIR >a:\report\mylist.doc
```

if the file you want to redirect output to is not on the current drive.

You can also redirect the `CHKDSK` command's disk status report to a file:

```
C>CHKDSK a: >status
```

Appending output

You can append the output of a command to the *end* of a file, using the redirection symbols `>>`. For example to add a file **index.dat** to the end of an existing file **book.txt**, type:

```
C>TYPE index.dat>>book.txt
```

The file **book.txt** now contains the contents of both files.

Redirecting input

Sometimes it is useful to make the contents of a file the input for a command. You achieve this using the redirection symbol `<`.

For example, to make the input to the `SORT` command (which sorts the contents of a file into alphabetical order) a file called **teleph.doc**, type:

C>SORT <teleph.doc

SORT takes the file, sorts its contents alphabetically and outputs the result to the screen.

Taking this a step further, to store the output of SORT into a file instead of just displaying it on screen, type:

C>SORT <teleph.doc>telelist.doc

SORT takes the **teleph.doc** file, sorts it into alphabetical order, and stores the result in the file **telelist.doc**.

Redirecting to or from a device

You can redirect input and output from and to a device rather than a file or command. Typical devices are printers, plotters and modems.

Device names

Devices are often named according to the communications port on the computer to which they are connected. Printers can be called PRN or LPT n , (where n is 1, 2 or 3). Table 12-1 lists the device names and shows their use with redirection.

Table 12-1 Device Names and Their Use With Redirection

Device	Description	Use
CON	User console (the keyboard and the screen).	Input and Output
AUX or COM1	First serial communications port.	Input and Output
COM2	Second serial port.	Input and Output
PRN or LPT1	First parallel printer.	Output
LPT2	Second parallel printer.	Output
LPT3	Third parallel printer.	Output
NUL	All output sent to this device is discarded.	Output

For details on how to set up a port for a particular device, refer to the following chapter, “Printing.”

12.2 Using pipes and filters

Pipes

Sometimes you will find it convenient to use the output from one program as the input to another. For example, the DIR command often produces a listing so long it cannot all be shown on the screen. In this case you can connect the output of DIR to the input of the MORE command, which displays output one screenful at a time. You can make this happen using a pipe; the piping symbol is | (a vertical bar).

This is how you would invoke the DIR and MORE commands together:

```
A>DIR|MORE
```

You can pipe together as many commands as you like. The commands you usually use with piping are known as filters (MORE is a filter command). Filters are further described in the next section.

Note

When you pipe programs together, the data piped between them are stored in temporary files created on the default disk; there must be sufficient space on the disk for them and the disk should not be write-protected. (The temporary files are removed when the operation is complete.)

Filter commands

Several commands act as filters and can be piped together. A filter command reads your input, acts upon it in some way and then outputs the result, usually to the screen. The main filter commands are:

Command	Function
Sort	Arranges data into alphabetical order before displaying it on the screen.
More	Displays a file's contents a screenful at a time.
Find	Extracts specified items of data for display.

For example, **Sort** is often used with the **Dir** command to sort the listing produced for example by filename or by size of file. Typing:

```
C>Dir | Sort /R
```

sorts the directory listing produced by **Dir** in reverse alphabetical order (the **/R** switch indicates it should be done from last to first instead of the default which is first to last).

The following command sorts the file **namelist.doc** using **Sort**. **Sort** arranges the file contents into alphabetical order before it is displayed on the screen:

```
C>Sort <namelist.doc
```

So, if the file **namelist.doc** contained the following names:

Henry
Sally
James
Sue

they would be arranged into alphabetical order as follows:

```
henry  
james  
sally  
sue
```

The following command sorts the file **tele.lst** into alphabetical order, and displays the result a screenful at a time:

```
C>Sort<tele.lst | More
```

In the following command line DIR and FIND are piped together to search for (FIND) and list (DIR) a directory called **ian**:

```
C>DIR|FIND ian
```

```
C>find ian
```

```
ian          <DIR>          3-01-90          3:42p
```

Note

For more details on these commands, see Chapter 11, “Commands.”

Combining pipes and redirection

You can also use piping and redirection together. For example:

```
C>DIR|SORT >dirtsort
```

produces a directory listing which is sorted into alphabetical order, and then stored in the file **dirtsort**.

Printing

YOUR system is able to support a large number of printers in a single configuration. It allows different users to all print simultaneously, and indeed lets individual users print on more than one printer at a time. This chapter describes the operations necessary to exploit these multiuser, multitasking print capabilities. The following topics are covered:

- ❑ Using the PRINTMAP command to associate the printers on your system with the logical device names understood by your applications.
- ❑ Running the print spooler using the CSPOOL utility, and if necessary reconfiguring the spooler for your particular environment by editing the **config.cpl** file.
- ❑ Using the MODE command to set parameters, such as baud rate and parity, for your serial ports.
- ❑ Using the PRINT command to send files to the printer.

13.1 Using PRINTMAP

Format

```
PRINTMAP [/H] [LPTn=PRN[n] | NUL] ... [COMn=AUX[n] | NUL] ...  
[/A] [/P] [/*]
```

Explanation

The operating system identifies the printer and serial ports on your system with the logical device names PRN and AUX, followed in each case by numeric values in the range 0 through 65. Most applications, however, understand the more limited range of device names available under DOS, namely LPT1, LPT2 and LPT3 for printer ports and COM1 and COM2 for serial ports.

Using the PRINTMAP command, you can map the printer and serial ports on your system to the logical names understood by your applications. You can display the current settings for each of your ports, and add explanatory text to the display to further identify the printers you have attached.

PRINTMAP settings apply to the current DOS session only, unless the /* switch is used. So you can have a different mapping of printers to device names for each of your DOS sessions, if you so wish.

To change the assignment of your printer ports use the following format:

```
PRINTMAP LPTn = PRN[n] | NUL
```

To change the assignment of your serial ports use the following format:

```
PRINTMAP COMn = AUX[n] | NUL
```

Use the NUL device to discard output from the specified port.

You can use both sets of parameters on a single command line, to map settings for both parallel and serial ports.

Use the command without any parameters to display the current settings, and any explanatory text that is available.

Command switches

/A	Displays a list of available auxiliary devices.
/P	Displays a list of available printers.
/*	Lets you change your assignments for all the virtual consoles on your terminal at once.

Remarks

The description of your port settings that displays when you invoke the PRINTMAP command with no parameters comes from a file called **printmap.dat**. This file is located in the same directory as the **printmap.exe** file.

The printmap.dat file

A **printmap.dat** file containing default descriptions for each possible printer and auxiliary device is supplied with the operating system. Edit this file, using any text editor, to provide yourself with a meaningful description of your various devices.

A sample description is as follows:

```
Printer 0 Parallel Port 0
Printer 1 Parallel Port 1
Printer 2 Parallel Port 2
Printer 3 COM1 Port
Printer 4 COM2 Port/Multiport 1
Printer 5 Multiport 2
Printer 6 Multiport 3
Printer 7 Multiport 4

Aux 0 COM1 Port
Aux 1 COM2 Port/Multiport 1
Aux 2 Multiport 2
Aux 3 Multiport 3
Aux 4 Multiport 4
```


Examples

To map printer port 2 to LPT1 and serial port 0 to COM2, and make this setting valid for all DOS sessions on your terminal, type:

```
C>PRINTMAP LPT1=PRN2 COM2=AUX0 /*
```

To display the current settings for your current DOS session, type:

```
C>PRINTMAP
```

You see a display similar to the following:

LPT1	Printer	0	Parallel Port	0
LPT2	Printer	1	Parallel Port	1
LPT3	Printer	2	Parallel Port	2
COM1	Aux	0	COM1 Port	
COM2	Aux	1	COM2 Port/Multiport	1

13.2 Running the CSPOOL print spooler

Often on multiuser computer systems, only one user can use a particular printer at any one time. Once a printer has been assigned to an application program, the printer can only be used by that application until the process is terminated or the application explicitly detaches from the printer.

This state of affairs can be frustrating for other users of the system, especially as few application programs inform the operating system when they have completed printing.

This operating system offers a way round these problems in the shape of the print spooler, CSPOOL. On a system where CSPOOL is installed, all printers are owned by the print spooler exclusively, unless configured otherwise. When an application prints a file, the file is intercepted by the print spooler and redirected into a spool file. Once the application has completed printing, the spool file is closed and the despool module sends the contents of the spool file to the required printer.

CSPOOL allows multiple spool files to be open at the same time, even if the files are to be printed at the same printer. Therefore, several applications can print to the same printer

simultaneously. A single user can also print to as many different printers as are configured in his system, again simultaneously.

The print spooler looks for one of the following conditions in order to detect that a print job is completed and so close the spool file:

- ☐ The application performs an explicit printer detach command.
- ☐ The process is terminated.
- ☐ The application does not send any characters to the printer for a certain time, known as a time-out period.

The time-out method is used with most DOS applications running under the operating system, since DOS applications rarely inform the operating system when they have finished printing.

You can configure the CSPOOL print spooler for your particular environment, using configuration parameters specified in the **config.cpl** file.

Installing CSPOOL

If CSPOOL forms part of your system, it should already have been installed using INSTALL at system installation time. It can also be installed at a later date, using SETUP (see Chapter 16, "Customizing your system").

Check that the SET command has been used to set the CSPOOL environment variable. The following command should be in the **cconfig.sys** file for the main console:

```
SET CSPOOL=c:\osutils\spool
```

If it is not there, add it now, then reboot your system and type:

```
C>SET
```

to check that the CSPOOL= entry is correct.

To verify that CSPOOL is now present, type:

```
C>CSPOOL
```

An introductory screen appears, offering a list of the various commands available with the CSPOOL utility. These commands are described in the following section.

CSPOOL command options

Only one copy of CSPOOL is required to provide print spooling for all users in a multiuser system. The print spooler provides a number of command options to manage print jobs. These options affect all users on the system, so you should take care not to disturb the work of other users when you use them.

The following command options are available:

Table 13-1 CSPOOL Command Options

Option	Explanation
START	Starts the print spooler.
QUIT	Removes the spooler from the system.
STATUS	Lists print jobs and gives their current state.
PRIORITY	Changes the priority of a queued job.
STOP	Removes a print job from the spool list.

Each option must be preceded on the command line by the command CSPOOL. Certain options require further parameters, as described in the following sections.

CSPPOOL START

Format

CSPPOOL START

Explanation

START starts up the spooler. The command can be included in the **autoexec.bat** file, but should only be executed from the main console, DOS session 1.

When the operating system runs the **autoexec.bat** file, the first parameter it is passed is the number of the current DOS session. To ensure that you are running the CSPPOOL START command from DOS session 1 you should include the following statement in the **autoexec.bat** file:

```
IF "%1"=="1" CSPPOOL START
```

Note

Make sure that your printer is turned on and that the paper is properly aligned before starting the print spooler.

If any jobs are pending from a previous session, the print spooler starts printing these as soon as it has been started up.

CSPPOOL QUIT

Format

CSPPOOL QUIT

Explanation

QUIT removes the spooler from the system. You should always remove the spooler before turning off the system. That way, any pending jobs are saved correctly and open files are closed.

CSPOOL STATUS

Format

CSPOOL STATUS [C]

Explanation

STATUS displays a list of jobs and shows their current state. Jobs currently printing are described as active. Jobs waiting for a printer are described as queued.

A maximum of four printers can print simultaneously.

Type CSPOOL STATUS to display a single status report.

Type CSPOOL STATUS C to get a continuously updated report.

CSPOOL PRIORITY

Format

CSPOOL PRIORITY job# value

Explanation

PRIORITY allows the priority of a queued job to be changed. For *job#* you must specify the job number (as shown in the first column of the STATUS display). For *value* enter a number between 1 and 10 indicating the job's print priority. A low number causes a job to be printed before other jobs. A high number causes a job to be printed after other jobs.

Example

To set the priority of job number 4 to 2, type:

CSPOOL PRIORITY 4 2

CSPPOOL STOP

Format

CSPPOOL STOP [*job#*]

Explanation

STOP removes a print job from the active or queued list.

For *job#* you should specify the job number of the job you wish to remove.

If you type CSPPOOL STOP without the job number, the status list of jobs is displayed followed by a prompt for the number of the job you wish to remove.

Example

To delete job number 5, type:

CSPPOOL STOP 5

Configuring the print spooler

The **config.cpl** file contains information needed to run the print spooler. You may need to edit this information to configure the print spooler for your particular environment.

To edit **config.cpl** you must use an editor that allows non-document mode, such as the Multiuser DOS editor, Editor.

All entries are case sensitive. This means that the keywords you type must be in upper case.

Process names are usually in upper case. However, some processes may contain lower case characters, such as "Tmp."

The following entries in **config.cpl** may be edited:

Table 13-2 Editable Config.cpl Entries

Entry	Explanation
TIME_OUT	Sets the spooler's time-out period.
FORMFEED	Gives the ASCII character used to produce a form-feed at the end of each print job.
NOFEED_JOBS	Lists any print jobs that produce their own form-feed and so do not require one from the spooler.
DEFAULT_PRIORITY	Sets the default priority for print jobs.
JOB_PRIORITY	Sets the priority for the named print jobs.
NOT_PRINTER	Lists any serial ports that are used not as printer but as auxiliary ports. Also identifies printers that are for private use and so not to be spooled.
CLEAR_SCREEN	Specifies the escape sequence used to clear the screen.
HOME_CURSOR	Specifies the escape sequence used to home the cursor.

These entries, and the parameters they take, are described in greater detail in the sections that follow.

Caution	Do not attempt to edit the other entries in the config.cpl file. Doing so may cause unpredictable results.
----------------	---

TIME_OUT

Format

TIME_OUT *n*

Explanation

TIME_OUT sets the time that needs to elapse without any characters being sent to the printer from an application, before the spooler closes the associated spool file and starts printing this job's data.

n is a decimal value in seconds. A sensible value is 15 seconds. Some applications, for example Supercalc4®, sending graphics to a printer, may pause for a long time between certain parts of a printout. If you notice that a particular job's printouts are split into several individual spool jobs, a higher TIME_OUT value is required.

FORMFEED

Format

FORMFEED *n*

Explanation

FORMFEED specifies the ASCII character used to indicate that the printer is to advance to the top of the next page. *n* is the decimal representation of the ASCII character. For most printers the value is 12. If you do not wish the spooler to add a form-feed after it has finished printing a job, enter the value 0 here.

NOFEED_JOBS

Format

```
NOFEED_JOBS job_name... END_LIST
```

Explanation

NOFEED_JOBS contains a list of jobs that do not require a form-feed. Several applications, such as WordStar®, SuperCalc®, and Xerox® Ventura Publisher™, send a form-feed to the printer themselves. The print spooler also sends a form-feed after completing the print job unless the name of the application is in the NOFEED_JOBS list. To avoid a blank page after the print-out, add the job name to this list.

The name of the job is usually the primary file name. The list must be terminated with the word END_LIST.

DEFAULT_PRIORITY

Format

```
DEFAULT_PRIORITY n
```

Explanation

DEFAULT_PRIORITY gives the priority assigned to jobs not listed in the JOB_PRIORITY list. When a new print job enters the spooler it is assigned a priority. Jobs with higher priorities are printed ahead of jobs with a lower priorities. 1 is the highest priority; 10 is the lowest priority. A suitable value for DEFAULT_PRIORITY is 5.

JOB_PRIORITY

Format

```
JOB_PRIORITY job_name n... END_LIST
```

Explanation

JOB_PRIORITY gives a list of jobs and the priorities to be assigned to them. Each entry has the format *job_name n*, where *n* is the priority number between 1 and 10. A lower number represents a higher priority.

The name of the job is usually the primary file name. The list must be terminated with the word `END_LIST`.

Example

The following example assigns priority 3 to the WS application, priority 2 to the VP application, and priority 1 to the SC application:

```
JOB_PRIORITY
WS 3
VP 2
SC 1
END_LIST
```

NOT_PRINTER

Format

```
NOT_PRINTER printer#... END_LIST
```

Explanation

`NOT_PRINTER` contains a list of the serial ports that are not to be used for printing but rather as auxiliary ports.

On most systems the serial ports may be used as printers or as auxiliary ports, for example for modems. The print spooler establishes ownership of all legal printer ports whose printer numbers are not in the `NOT_PRINTER` list.

You must create a `NOT_PRINTER` list if you want to use a program such as a modem communications program, which establishes auxiliary port ownership when it runs.

Use `NOT_PRINTER` in addition when you wish to identify a printer as private, and therefore not to be spooled.

The list must be terminated with the word `END_LIST`.

Example

If COM1 and COM2 (AUX 0 and AUX1) are to be used as modem ports, rather than as printers, you should list printer numbers 3 and 4, which correspond to the AUX 0 and AUX 1 serial ports, in the `NOT_PRINTER` list as follows:

```
NOT_PRINTER 3 4 END_LIST
```

CLEAR_SCREEN

Format

`CLEAR_SCREEN escape_sequence 0`

Explanation

`CLEAR_SCREEN` specifies the escape sequence used to clear the screen. The entry must be in decimal. Normally this is set to 27 69, the sequence used by the operating system on the main console.

The last entry in the list must be 0.

HOME_CURSOR

Format

`HOME_CURSOR escape_sequence 0`

Explanation

`HOME_CURSOR` specifies the escape sequence used to home the cursor. The entry must be in decimal. Normally this is set to 27 69, the sequence used by the operating system on the main console.

The last entry in the list must be 0.

Configuring DOS applications for the print spooler

Operating-system-aware applications are specifically intended for multitasking and multiuser environments, and need no special configuration for use with the print spooler. DOS applications, however, often go to hardware directly, instead of allowing the operating system to manage these resources. For DOS applications to run with the print spooler, they must be configured to use system services, rather than hardware services, to “talk” to the printers.

The ideal situation for DOS applications is for them to use the PRN or LST device. However, most DOS applications use INT 17 calls on ports LPT1 and LPT2. These applications will work correctly with the print spooler but applications configured for the COM ports may produce unpredictable results.

If your application does not support PRN or LST devices, configure the application to use LPT1 or LPT2. If your system does not have a parallel printer (LPT1 or LPT2) you can still configure your application to use LPT1 or LPT2 and use PRINTMAP to map LPT1 or LPT2 to printer 3 or 4 (the serial printers on a standard system).

If your DOS application does not print correctly with the spooler installed, check the configuration. If necessary, request support from the application vendor on how to configure the application to use the PRN, LST, or LPT devices.

Hints on using CSPOOL

Normally, you will not be aware of the spooler, as most functions are managed internally. When you first use the spooler, or when moving new applications on to the system, make sure that time-outs and form-feeds are correct and adjust the **config.cpl** file if necessary.

A print spooler is not a printer buffer, so print jobs are not necessarily performed faster with the print spooler than without it. Usually, control of your application is returned to you sooner when the print spooler is installed. However, the time taken to complete the printing may be longer with the spooler installed. This is because the print spooler only starts printing when the application has finished sending characters to the printer. Thus there is a time-lag between the completion of a print job from the application's point of view and from the printer's point of view.

In an environment where regular print jobs are performed from multiple consoles the benefits of the spooler are very noticeable, since it is no longer necessary to ask other operators to exit their applications in order to free the printer.

The perceived performance of the spooler also depends on the type of printer being used. A dot matrix printer usually locks applications for extended periods. With the print spooler, however, control is returned to the application relatively fast. On the other hand, laser printers are likely to take a longer time to complete when the print spooler is installed. The large amount of data first needs to be spooled to the disk and then from the disk to the printer.

Since the spooler owns the printer as a resource, you cannot use the **Ctrl+P** command to attach the printer to the console. However, you may use command line redirection to print. For example, to send the directory to the printer, type:

DIR > PRN:

As the spooler may have files open at any time after any print activity occurs on the system, you should terminate the spooler before switching off the system. This ensures that no open files are on the system. You may also reboot the system when all applications have been exited and all print jobs have completed.

Use the **CHKDSK /F** command regularly to convert any lost clusters created by switching off the system while there are open files. Be sure to make back ups before doing this..

Do not delete any files in the **SPOOL** directory while the print spooler is running. If you wish to delete files in the **SPOOL** directory you must first use the print spooler **QUIT** to terminate the spooler. You may then remove all **.spw**, **.spr**, and **.spl** files. Also, do not attempt to access the files in the spool directory while the print spooler is active. This could produce unpredictable results.

13.3 Using MODE

MODE is a multi-faceted command that allows you to perform a number of hardware configuration operations. As far as printing goes, there are three relevant forms of the **MODE** command. These allow you to:

- ☐ Set up a printer.
- ☐ Set up a serial port.
- ☐ Assign a printer.

Setting up a printer

Format

MODE LPT#[:][n][, [m]]

Explanation

Use this format of MODE to set up a printer connected to a particular port. Your printer documentation should help indicate the values you need (for example for *n*).

Option	Explanation
LPT#:	The printer number, where # can usually be 1, 2, or 3.
<i>n</i>	The printer characters per line: 80 or 132. The default is 80. (80 character printers are often known as “narrow carriage”, and 132 character printers as “wide carriage”.)
<i>m</i>	The lines per inch for the printer: 6 or 8 (8 is the default).

Examples

This command sets up a 132-column wide printer at printer port LPT2.

```
C>MODE LPT2:132
```

The following command returns to the default settings on the same port:

```
C>MODE LPT2
```

Setting up a serial port

Format

```
MODE COM#[:]baudrate[, [parity], [databits], [stopbits]]
```

Explanation

This form of the MODE command enables you to set parameters for your serial (or communication) ports, for example COM1 and COM2. It operates on the logical device names mapped to physical devices through the PRINTMAP command.

The command allows you to set baud rate, parity, data bits, and stop bits.

Usually you will find that you only need to set the baud rate; the default values for the other options may already be set up for your needs.

Options	Explanation
<i>baudrate</i>	Can be 110, 150, 300, 600, 1200, 2400, 4800, 9600, 19200, 38400, 56000 or 115200. You only need to type the first two digits of your selection.
<i>parity</i>	Can be N (none), O (odd) or E (even, which is the default).
<i>databits</i>	Can be 7 or 8 (7 is the default).
<i>stopbits</i>	Can be 1 (the normal default) or 2 (the default when the baud rate is set at 110).

Note

Always separate the different options in the command line with commas. If options are not required, the correct number of commas must still be inserted.

Examples

The following command sets the communications parameters (sometimes called protocol) on COM1 to 9600 baud. The defaults for the other values will be: even parity, 7 databits, 1 stopbit.

```
C>MODE COM1:96
```

The following command changes the protocol for the above command to odd parity:

```
C>MODE COM1:96,o
```

The following command sets the protocol on COM2 at 4800 baud and 8 databits:

```
C>MODE COM2:48,,8
```

Assigning a printer

Format

```
MODE LPT#[:]=COM#
```

Explanation

With this form of MODE you can alter the standard assignments for your printer ports. It is offered for DOS compatibility, and only maps one logical device name to another. Physical mapping is achieved by means of the PRINTMAP utility.

Usually, information is sent to print through your computer's parallel ports, commonly labeled LPT1, LPT2, and LPT3. The MODE command enables you to send information to a serial printer through one of your computer's serial ports, COM1 or COM2, instead. For example if you have a program (such as a word processor) that normally sends output to port LPT1, you can use MODE to redirect information to a serial port like COM1 instead.

The hash symbol # represents the number of the port, for example LPT# could be LPT1, LPT2 or LPT3. The colon between LPT# and =COM# is optional.

To return to the default settings, type the LPT#: without the COM#.

Examples

To redirect output from LPT1 to COM1, type the following:

```
C>MODE LPT1:=COM1
```

To return to the default settings type:

```
C>MODE LPT1:
```


13.4 Using PRINT

Once you have the correct port settings in place, you can begin to print files. Use the PRINT command for this.

PRINT

Format

```
PRINT [/H] [wildspec] [/B:buffsize] [/C] [/D:device]  
[/M:maxticks] [/P] [/Q:queuesize] [/S:timeslice] [/T]  
[/U:busyticks]
```

Explanation

PRINT is used to print text files.

When you have created files on disk that need printing, you can use PRINT to manage the print process. PRINT creates a list of files called a *print queue* and then prints the files one by one.

You can run PRINT in a background DOS session, leaving your computer free for you to carry on working as normal. PRINT is an external command, but part of it remains loaded in memory. When you start the PRINT command, you see the message:

```
Resident part of PRINT installed
```

Command switches

<i>/B:buffsize</i>	Sets the size of the print buffer (in bytes). The default is 512. A larger buffer can (sometimes) improve the print speed. The buffer size can only be set the first time the PRINT command is used during a working session.
<i>/C</i>	Deletes the named file(s) from the print queue. Filenames can be typed before or after this option in the command line: the option looks at the one filename immediately preceding it in the command line and at all the filenames typed after it.

Command switches (continued)

<i>/D:device</i>	Sends the printer output to the named device. The default is PRN (which is the same as LPT1). The other valid devices you can name are: AUX, COM1, COM2, LPT1, LPT2, and LPT3. The device name can only be specified the first time the PRINT command is used during a working session.
<i>/M:maxticks</i> <i>/S:timeslice</i> <i>/U:busyticks</i>	These parameters work together to govern what portion of computer time PRINT is allowed to use. <i>Timeslice</i> (default 8) describes how often the computer gives control to the PRINT utility; <i>maxticks</i> (default 2, range 1 through 255) governs for how many clock-ticks PRINT runs once it has control; <i>busyticks</i> (default 1) specifies how long PRINT waits before handing back control if the printer is busy. (A clock-tick is approximately 1/18th of a second.) Each of these parameters can only be specified the first time you issue the PRINT command during a working session.
<i>/P</i>	Adds the named files to the print queue. The rules for using it are the same as those governing the /C option.
<i>/Q:queuesize</i>	Sets the maximum number of files that can be held in the print queue. The default is 10, in the range 4 to 32. This parameter can only be defined the first time you use PRINT during a working session.
<i>/T</i>	Deletes all files from the print queue. The printer prints the message: All files cancelled by operator

Examples

If you type this command when using PRINT for the first time after starting up your system:

```
C>PRINT /Q:20 chap1.doc
```

the command sets the spool queue size to 20. As the /D option was not given, you get this message:

```
List device? [PRN]
```

You need to press **Enter** to print via the default port LPT1 (LPT1 is equivalent to PRN). Type the port name if you want to print via another port. If PRINT was already started, using the /Q option would result in an error message.

The following command deletes the files **chap2.doc** and **chap3.doc** from the print queue:

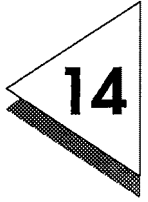
```
C>PRINT chap2.doc /C chap3.doc
```

PRINT allows you to combine several commands on one command line.

The following command deletes all the files on the print queue and then adds files **newbook1.doc** and **newbook2.doc** to the queue:

```
C>PRINT /T newbook1.doc newbook2.doc
```

The SID debugging tool



SID is a powerful debugger. It allows you to test and debug programs interactively.

Most users will not need to use SID, but a description is included in this guide for completeness.

14.1 SID basics

Starting SID

To start SID, type SID at the command prompt for example:

```
C>SID
```

You see the SID copyright message and prompt. SID is ready to accept your commands. You can now use the E command to load a program for a debugging session. SID tells you where in memory the program is loaded, for example:

```
#eprogram.exe  
  
      Start      End  
1000:0000 1000:2000  
  
#
```

Alternatively, you can type SID at the command prompt followed by the name of the program you want to debug, for example:

```
C>SID program.exe
```

Quitting SID

To quit SID type Q or **Ctrl+C** at the # prompt. This returns control to the operating system.

SID command conventions

A command line in SID can be up to 64 characters in length, and must be terminated with a carriage return. SID does not process the command line until you press **Enter**.

The first character, or in some cases the first two characters, of each command line determines the command action. The command character can be followed by one or more arguments. Arguments must be separated from each other by commas or spaces. No spaces are allowed between the command character and the first argument.

Note that if the first character of a SID command line is a semicolon (;), the entire line is treated as a comment and ignored.

Specifying a 20-bit address

Most SID commands require one or more addresses as operands. Because the 8086 processor can address up to 1 Mbyte of memory, addresses must be 20-bit values. Enter a 20-bit address as follows:

```
ssss:oooo
```

where *ssss* represents an optional 16-bit segment number and *oooo* is a 16-bit offset. SID combines these values to produce a 20-bit effective address:

```
ssss0
```

```
+oooo
```

```
---
```

```
eeee
```

The segment value, *ssss*, is optional. If you omit the segment value, a default value appropriate to the command being executed, is used.

Hexadecimal numbers

SID normally accepts and displays values in hexadecimal. The valid hexadecimal digits are the decimal digits 0 through 9 and the hexadecimal digits A, B, C, D, E, and F, corresponding to the decimal values 10 through 15.

Decimal numbers

To enter decimal numbers, precede the number with the # symbol. The number that follows must comprise one or more decimal digits (0 through 9).

Character values

SID accepts ASCII characters, enclosed in single quotation marks, as literal values in expressions. Characters remain as

typed within the paired quotation marks that is, no case translation occurs.

Operators in expressions

Numbers and strings can be combined into expressions using + and - operators. The entire sequence must be written without embedded blanks.

14.2 SID commands

SID commands give you control of program execution and allow you to display and modify system memory and the CPU state. This section lists the SID commands in alphabetical order. The commands are summarized in Table 14-1.

Table 14-1 Summary of SID Commands

Command	Explanation
?	Displays available commands.
??	Displays detailed command list.
A	Assembles 8086 mnemonics directly into memory.
B	Block compares memory and displays discrepancies found.
D	Displays memory in byte or word format.
E	Loads a file into memory so that the next G, T or U command can begin program execution.
F	Fills an area of memory with a byte or word constant.
G	Transfers control to the program being tested and optionally sets one or two breakpoints.
H	Provides hexadecimal arithmetic functions.
<i>Istring</i>	Sets up command line arguments.
L	Lists the contents of memory in assembly language.
M	Copies a block of data values from one area of memory to another.
P	Sets, clears and lists current breakpoints.
Q	Exits from the SID program.
QI, QO	Allows access to any of the 65,536 input/output ports. QI reads data from a port and QO writes data to a port.
QR, QW	Enables you to read and write data from absolute locations on a given disk drive.

Table 14-1 Summary of SID Commands (continued)

Command	Explanation
R	Reads a file into a contiguous block of memory.
S	Changes the contents of bytes or words of memory.
SR	Searches for a given ASCII “string” or pattern of bytes and lists the addresses where the pattern occurs.
T	Executes one or more instructions.
U	This is the same as T, but without register display.
V	Displays the values of the last file loaded using the E or R commands.
W	Writes the contents of a contiguous block of memory to disk file.
X	Examines or modifies the CPU registers and flags.
Z	Displays registers of an attached 8087 maths coprocessor.

? & ?? (Help)

The ? command prints a list of available SID commands. The format is as follows:

?

The ?? command prints a detailed command list that, in addition to the SID commands, includes the available command options. The format is as follows:

??

A (ASSEMBLE)

The A command assembles 8086 mnemonics directly into memory. It has the format:

As

where *s* is the 20-bit address at which assembly starts.

SID responds to the A command by displaying the address of the memory location where assembly begins. At this point, you enter assembly language statements. When you enter each statement, SID converts it to binary, places the value(s) in memory, and displays the address of the next available memory location. This process continues until you enter a blank line or a line containing only a period. SID responds to invalid statements by displaying a question mark, ?, and redisplaying the current assembly address.

Note

You can use an expression wherever a numeric value is valid in an assembly language statement. Under the A command, references to registers refer to the names of the registers, not to the contents of the registers as in other SID commands. Under the A command, there is no way to reference the contents of a register in an expression.

Examples

To assemble at 4000:213:

```
#a4000:213
```

To set AX register to decimal 128:

```
4000:0213 mov ax,#128
```

To push AX register on stack:

```
4000:0216 push ax
```

B (BLOCK COMPARE)

The B command compares two blocks of memory and displays any discrepancies at the screen. The B command has the format:

`Bs1, f1, s2`

where *s1* is the 20-bit address of the start of the first block; *f1* is the offset of the final byte of the first block, and *s2* is the 20-bit address of the start of the second block.

If the segment is not specified in *s2*, the same value is used that was used for *s1*.

Any differences in the two blocks are displayed on the screen in the format:

`a1 b1 a2 b2`

where *a1* and the *a2* are the addresses in the blocks; *b1* and *b2* are the values at the indicated addresses. If no differences are displayed, the blocks are identical.

Example

To compare 200H bytes starting at 40:0 with the block starting at 60:0:

`#b40:0,1ff,60:0`

D (DISPLAY)

The D command displays the contents of memory as 8-bit or 16-bit hexadecimal values, and in ASCII. The command has the following formats, where *s* is the 20-bit address where the display starts, and *f* is the 16-bit offset within the segment specified in *s* where the display finishes:

- (a) D
- (b) Ds
- (c) Ds, f
- (d) DW
- (e) DWs
- (f) DWs, f

Memory is displayed on one or more display lines. Each display line shows the values of up to 16 memory locations.

Format (a) displays memory from the current display address for 12 display lines. Response to (b) is similar to format (a) but the display address is first set to be the 20-bit address *s*.

Format (c) displays the memory block between locations *s* and *f*.

The display line appears as follows:

```
.  ssss:oooo bb bb . . . bb cc . . . c
```

where *ssss* is the segment being displayed, and *oooo* is the offset within segment *ssss*. The *bb*'s represent the contents of the memory locations in hexadecimal, and the *c*'s represent the contents of memory in ASCII. A period represents any non-graphic ASCII character.

Formats (d), (e) and (f) are analogous to the first three formats, except that the contents of memory are displayed as 16-bit values, rather than 8-bit values, as shown below:

```
ssss:oooo www www . . . www cccc . . . cc
```

During a long display, you can abort the D command by typing **Ctrl+Break** at the console.

Example

To display memory bytes from offset F00H through F23H in the current data segment:

```
#df00,f23
```

E (LOAD PROGRAM)

The **E** command loads a file into memory so that the next **G**, **T**, or **U** command can begin program execution. The **E** command has the formats:

(a) **E** filename

(b) **E**

Format (a) loads the file by the name *filename*. The file is assumed to be either a **.com** or **.exe** file. If no file type is specified, **.exe** is assumed. To load a **.com** file, you must specify the file type with the filename. The contents of the **CS**, **DS**, **ES**, **SS**, **SP**, and **IP** registers are altered according to the type of file loaded. When the load is complete, **SID** displays the start and end address of the memory block where the file was loaded. Use the **V** command to redisplay this information later.

Format (b) releases all memory being used by **SID** for programs.

When loading a program file with the **E** command, **SID** releases any blocks of memory used by previous **E** or **R** commands, or by programs executed under **SID**. Only one file can be loaded for execution at a time.

SID issues an error message if a file does not exist or cannot be successfully loaded in the available memory.

Example

To load file **test.exe**:

```
#etest
```

F (FILL)

The **F** command fills an area of memory with a byte or word constant. The following are possible formats:

(a) **Fs, f, b**

(b) **FWs, f, w**

The *s* is a 20-bit starting address of the block to be filled, and *f* is a 16-bit offset of the final byte of the block within the segment specified in *s*.

In response to form (a), SID stores the 8-bit value *b* in locations *s* through *f*. In form (b), the 16-bit value *w* is stored in locations *s* through *f* in standard form, low 8-bits first followed by high 8-bits.

If *s* is greater than *f* or the value *b* is greater than 255, SID responds with a question mark. SID issues an error message if the value stored in memory cannot be read back successfully, indicating faulty or non-existent RAM at the location indicated.

Example

To fill memory from 100H through 13FH with 0:

#f100,13f,0

G (GO)

The G command transfers control to the program being tested, and optionally sets one or two breakpoints. The following are possible formats:

- (a) G
- (b) G, b1
- (c) G, b1, b2
- (d) Gs
- (e) Gs, b1
- (f) Gs, b1, b2
- (g) -G [s|s, b1|s, b1, b2]

where *s* is a 20-bit address where program execution is to start, and *b1* and *b2* are 20-bit addresses of breakpoints.

In formats (a), (b), and (c), no starting address is specified, so SID derives the 20-bit address from the CS and IP registers. Format (a) transfers control to your program without setting any breakpoints. Formats (b) and (c) set one and two breakpoints respectively before passing control to your program. Formats (d), (e), and (f) are analogous to (a), (b), and (c), except that the CS and IP registers are first set to *s*. The formats in (g) are analogous to formats (a) through (f), except that intermediate pass point displays are suppressed.

Once control has been transferred to the program under test, it executes in real time until it encounters a breakpoint. SID then regains control, clears the breakpoints set by the G command, and indicates the address at which execution of the program under test was interrupted, as shown:

```
*SSSS:0000
```

The *ssss* corresponds to the CS, *0000* corresponds to the IP where the break occurred. When a breakpoint returns control to SID, the instruction at the breakpoint address has not yet been executed.

Example

To begin execution at default address with no breakpoints set:

```
#g
```

H (HEXADECIMAL MATH)

The H command provides several useful arithmetic functions. The possible formats are as follows:

(a) Ha,b

(b) Ha

Format (a) computes the sum (*ssss*), difference (*dddd*), product (*pppppppp*), and quotient (*qqqq*) with the remainder (*rrrr*) of two 16-bit values. The results are displayed in hexadecimal as below:

+ ssss - dddd * pppppppp / qqqq (rrrr)

Underflow and overflow are ignored in addition and subtraction.

Format (b) displays the value of the expression *a* in hexadecimal, decimal, ASCII (if the value has a graphic ASCII equivalent), forms as shown:

hhhh #dddd 'c'

You can abort the display by pressing any key at the console.

Example

To show sum, difference, product, and quotient of 5C28H and 80H:

#h5c28,80

.

I (INPUT COMMAND TAIL)

The I command prepares a file control block and command tail buffer in SID's program segment prefix, and copies this information into the program segment prefix of the last file loaded with the E command. The command has the following format:

`Icommand_tail`

The *command_tail* is a character string that usually contains one or more filenames. The first filename is parsed into the default file control block at 005CH. The optional second filename is parsed into the second part of the default file control block beginning at 006CH. The characters in *command_tail* are also copied into the default command buffer at 0080H. The length of *command_tail* is stored at 0080H, followed by the character string terminated with a binary zero.

If a file has been loaded with the E command, SID copies the file control block and command buffer from the program segment prefix of SID to the program segment prefix of the program loaded. The location of the program segment prefix of a program loaded with the E command is the value displayed for DS on completion of the program load.

Examples

To set up a file control block at 5CH for the file **file1.exe** and put the string "file1.exe" in the buffer at 80H (in the program segment prefix of the last file loaded with an E command):

```
#ifile1.exe
```

To set up file control blocks at 5CH and 6CH for the files **a:file1** and **b:file2**, and copy the string following the "i" into the buffer at 80H:

```
#ia:file1 b:file2 c:file3 $px
```

L (LIST)

The L command lists the contents of memory in assembly language. The following are possible formats:

- (a) L
- (b) Ls
- (c) Ls, f

The *s* is a 20-bit address where the list starts, and *f* is a 16-bit offset within the segment specified in *s* where the list finishes.

Each disassembled instruction is in the format:

ssss:oooo *prefixes* *opcode* *operands*

where *prefixes* are segment override, lock and repeat prefixes, *opcode* is the mnemonic for the instruction, and the *operands* field contains 0, 1, or 2 operands, as required by the instruction.

Format (a) lists twelve disassembled instructions from the current list address. Format (b) sets the list address to *s* and then lists twelve instructions. Format (c) lists disassembled code from *s* through *f*.

The list address is always set to the next unlisted location in preparation for the next L command. When SID regains control from a program being tested (see G, T, and U commands), the list address is set to the current value of the CS and IP registers.

You can abort long displays by typing **Ctrl+Break** during the list process or by entering **Ctrl+S** to halt the display temporarily. **Ctrl+Q** will restart the display.

If the memory location being disassembled is not a valid 8086 instruction, SID displays it in the form:

??=nn

The *nn* is the hexadecimal value of the contents of the memory location.

Example

To disassemble instructions from 243CH through 244EH:

#1243c,244e

M (MOVE)

The M command moves a block of data values from one area of memory to another. The M command has the format:

Ms,f,d

The *s* is the 20-bit starting address of the block to be moved, *f* is the offset of the final byte to be moved within the segment described by *s*, and *d* is the 20-bit address of the first byte of the area to receive the data. If the segment is not specified in *d*, the same value is used as was used for *s*.

Note

If *d* is between *s* and *f*, part of the block being moved is overwritten before it is moved, because data is transferred starting from location *s*.

Example

To move 10 bytes from 20:2400 to 30:100:

#m20:2400,+9,30:100

P (PASS POINT)

The P command sets, clears, and displays pass points. The following are the possible formats:

- (a) Pa,n
- (b) Pa
- (c) -Pa
- (d) -P
- (e) P

A pass point is a permanent breakpoint that remains in effect until it is explicitly removed, as opposed to breakpoints set with the G command, which must be reentered with each G command. Pass points have associated pass counts ranging from 1 to 0FFFFH. The pass count indicates how many times the instruction at the pass point executes before the control returns to the console. Up to sixteen pass points can be set at a time.

An important distinction between breakpoints and pass points is that when execution stops at a breakpoint, the instruction at the breakpoint has not yet been executed. When execution stops at a pass point whose pass count has reached 1, the instruction at the pass point has been executed. This makes it simple to proceed from a pass point with a G command without immediately encountering the same pass point.

Formats (a) and (b) are used to set pass points. Format (a) sets a pass point at address *a* with a pass count of *n*, where *a* is the 20-bit address of the pass point, and *n* is the pass count, in the range 1 to 0FFFFH. If a pass point is already active at *a*, the pass count is simply changed to *n*. SID responds with a question mark if there are already 16 active pass points.

Format (b) sets a pass point at *a* with a pass count of 1. If a pass point is already active at *a*, the pass count is simply changed to 1. SID responds with a question mark if there are already 16 active pass points. Formats (c) and (d) are used to clear pass points. Format (c) clears the pass point at location *a*. SID responds with a question mark if there is no pass point set at *a*. Format (d) clears all the pass points. Format (e) displays all the active pass points in the form:

```
nnnn ssss:oooo
```

where *nnnn* is the current pass count for the pass point, *ssss:0000* is the segment and offset of the pass point location. When a pass point is encountered, SID displays the pass point information in the format:

```
nnnn PASS ssss:0000
```

Next, SID displays the CPU state before the instruction at the pass point is executed. SID then executes the instruction at the pass point. If the pass count is greater than 1, SID decrements the pass count and transfers control back to the program under test. When the pass count reaches 1, SID displays the break address (that of the next instruction to be executed) in the following format:

```
*ssss:0000
```

Once the pass count reaches 1, it remains at 1 until the pass point is cleared or the pass count is changed with another P command.

Use pass points with the G, T, and U commands. When you use the G or U command, you can suppress the intermediate pass point display with the -G or -U formats (see the G and U commands). In this case, only the final pass points (when the pass count equals 1) are displayed. You can interrupt the G or U command before its normal termination by pressing **Ctrl+Break** at the console. SID aborts the G or U command when the next pass point is encountered and prompts for the next command. You can also use pass points with breakpoints set with the G command.

If a pass point and a breakpoint are set at the same address, the breakpoint is encountered first. Otherwise, pass points behave in the usual manner, decrementing the pass count until it reaches 1, and then returning control to SID.

Normally, the segment registers are not displayed at pass points. You can use the S command to enable/disable the segment register display.

Example

To set a pass point at address 1000:1234:

```
#p1000:1234
```

Q (QUIT)

The Q command exits from SID.

QI, QO, QR, QW (QUERY I/O)

The QI and QO commands allow access to any of the 65,536 input/output ports. The QI command reads data from a port; the QO command writes data to a port. The QI command has the following formats:

- (a) QIn
- (b) QIWn

where n is the 16-bit port number. In the first case, SID displays the 8-bit value read from port n . In the second case, SID displays a 16-bit value from port n .

The QO command has the following formats:

- (c) QOn, v
- (d) QOWn, v

where n is the 16-bit port number, and v is the value to output. In the first case, the 8-bit value v is written to port n . If v is greater than 255, SID responds with a question mark. In the second case, the 16-bit value v is written to port n .

Examples

To display the 16-bit value of input port 20H:

```
#qiW20
```

To display the 8-bit value of input port 1024:

```
#qi#1024
```

To set the 16-bit output port number 20H to 0FF7EH:

```
#qow20,FF7E
```

To set the 8-bit output port number 1025 to 2:

```
#qo#1025,2
```


The QR and QW commands enable you to read and write data from absolute locations on a given disk drive. The two commands have identical formats:

(e) QR*s*,*drive*,*sector*,*count*

(f) QW*s*,*drive*,*sector*,*count*

where *s* is the segment:offset in memory into/from which data is to be read/written, *drive* is the drive number to use, for example 0=A:, 1=B:, *sector* is the starting sector number, and *count* is the number of sectors to process.

Examples

To read 2 sectors from logical sector 15 of drive A: into memory starting at 9000:0000:

QR9000:0,0,15,2

To write the same two sectors back out to drive B:

QW0,1,15,2

Notice that the full segment part of the address need not be given if it has not changed since the last command.

This command must be used with the upmost care, as vital information such as directories or file allocation tables can be damaged by incorrect use.

Note

A 32-bit sector number can be used which takes the form *nnnn:nnnn*, that is 0003:ffff for sector 3ffff.

R (READ)

The R command reads a file into a contiguous block of memory. The R command has the formats:

- (a) Rfilename
- (b) Rfilename, s

where *filename* is the name and type of the file to be read, and *s* is the location to which the file is read. Format (a) lets SID determine the memory location into which the file is read. Format (b) causes SID to read the file into the memory segment beginning at *s*. This address can have the standard form (*ssss:oooo*) as in the following example:

```
#RCOMMAND.COM,1000:0
```

The low-order four bits of *s* are assumed to be zero, so SID reads files on a paragraph boundary.

SID reads the file into memory and displays the start and end addresses of the block of memory the file occupies. A V command can redisplay this information later. The default display pointer (for further D commands) is set to the start of the block occupied by the file.

The R command does not free memory used by another R or E command. Thus, many files can be read into memory without overlapping.

SID issues an error message if the file does not exist or there is not enough memory to load the file.

Examples

To read file **sid.com** into memory.

```
#rsid.com
```

To read file **test** into memory.

```
#rtest
```

To read file **test** into memory starting at location 1000:0.

```
#rtest,1000:0
```

S (SET)

The S command changes the contents of bytes or words of memory. The following are possible formats:

- (a) Ss
- (b) SWs
- (c) S
- (d) -S

where *s* is the 20-bit address where the change occurs.

SID displays the memory address and its current contents on the following line. In response to format (a), the display is

```
ssss:0000 bb
```

and in response to format (b), the display is

```
ssss:0000 wwww
```

where *bb* and *wwww* are the contents of memory in byte and word formats, respectively.

In response to one of these displays, you can choose to alter the memory location or to leave it unchanged. If you enter a valid expression, the contents of the byte (or word) in memory is replaced with the value of the expression. If you do not enter a value, the contents of memory are unaffected and the contents of the next address are displayed. In either case, SID continues to display successive memory addresses and values until you enter a period on a line by itself, or until SID detects an invalid expression.

In response to format (a), you can enter a string of ASCII characters, beginning with a quotation mark and ending with a carriage return. The characters between the quotation mark and the carriage return are placed in memory starting at the address displayed. No case conversion takes place. The next address displayed is the address following the character string. SID issues an error message if the value stored in memory cannot be read back successfully, indicating faulty or non-existent RAM at the location indicated.

Formats (c) and (d) control the display of the segment registers when the CPU state is displayed with the trace command and at pass points. Format (c) turns on the segment register display, and format (d) turns it off. It is often convenient to turn

off the segment register display while debugging to allow the CPU state display to fit on one line.

Examples

To set byte to 0:

```
1000:1234 55 0
```

To set 3 bytes to 'a', 'b', 'c':

```
1000:1235 55 "abc
```

To set byte to decimal 75:

```
1000:1238 55 #75
```

To terminate the set command:

```
1000:1239 55 .
```

To enable segment register display in CPU state display:

```
#s
```

To disable segment register display in CPU state display:

```
#-s
```

SR (SEARCH)

The SR command searches a block of memory for a given pattern of numeric or ASCII values, and lists the addresses where the pattern occurs. The SR command takes the form:

`SRs, f, pattern`

where *s* is the 20-bit starting address of the block to be searched, *f* is the offset of the final address of the block, and *pattern* is a list of one or more hexadecimal values and/or ASCII strings. ASCII strings are enclosed in double or single quotation marks and can be any length.

For each occurrence of *pattern*, SID displays the 20-bit address of the first byte of the pattern, in the form:

`ssss:oooo`

If no addresses are listed, the pattern was not found.

Example

To search the memory block between 0:0 and 0:0FFFFH for the 3- character pattern that starts with “x”, ends with “y”, and has a decimal 27 in the middle:

`# sr0:0,ffff,“x”,#27,“y”`

T (TRACE)

The T command traces program execution for 1 to 0FFFFH program steps, displaying the CPU state before each step. The T command takes the following forms:

- (a) T
- (b) T_n
- (c) TW
- (d) TW_n

The *n* is the number of program steps to execute before returning control to the console. If *n* is omitted, a single program step is executed. A program step is generally a single instruction, with the following exceptions:

- ❑ If an operating system interrupt instruction is traced, the entire function is treated as one program step, and executes in real time. This is because SID itself makes operating system calls, and the operating system is not reentrant.
- ❑ The 8086 has a feature that disables interrupts (including the Trace Interrupt) for one instruction after a MOV or POP loads a segment register. For this reason, if the traced instruction is a MOV or POP whose destination is a segment register, the CPU executes the next instruction immediately. This prevents an interrupt occurring when the stack is undefined, for example between the two instructions in a sequence such as:

```
MOV SS, STACKSEGMENT  
MOV SP, STACKOFFSET
```

A sequence of such MOV or POP instructions plus one instruction after the sequence is considered a one program step.

- ❑ If any of the TW forms are used and the traced instruction is a CALL, CALLF or INT, the entire called subroutine or interrupt handler (and any subroutines called therein) is treated as a one program step and executes in real time.

Before each program step is executed, SID displays the CPU state, the disassembled instruction to be executed, and the contents of the memory location(s) referenced by the instruction, if appropriate. (See the X command for a detailed description of the CPU state display.)

The segment registers are not normally displayed with the T command, which allows the entire CPU state to be displayed on one line. To enable the segment register display, use the S command (see the S command). With the segment register display enabled, the display of the CPU state is identical to that of the X command.

In all the forms, control transfers to the program under test at the address indicated by the CS and IP registers. If *n* is not specified, as in format (a), one program step is executed. Otherwise, SID executes *n* program steps and displays the CPU state before each step, as in format (b). You can abort a long trace before *n* steps have been executed by typing **Ctrl+Break** at the console.

When *n* steps have been executed, SID displays the address of the next instruction to be executed in the form:

```
*ssss:oooo
```

Formats (c) and (d) are analogous to formats (a) and (b), except in the way subroutine calls are treated. In the TW forms, the entire subroutine called from the program level being traced is treated as a single program step, and executes in real time. This allows tracing at a high level of the program, ignoring subroutines that have already been debugged, or for other reasons are not currently of interest.

When a single instruction is being traced, interrupts are disabled for the duration of the instruction. This prevents SID from tracing through interrupt handlers when debugging on systems in which interrupts occur frequently.

After a T command, the list address used in the L command is set to the address of the next instruction to be executed, and the default segment values are set to the CS and DS register values.

Examples

To trace one program step:

```
#t
```

To trace 65535 steps:

```
#tffff
```

U (UNTRACE)

The U command is similar to the T command except that the CPU state is displayed only before the first instruction is executed, rather than before every step. The following are possible formats:

- (a) U
- (b) Un
- (c) UW
- (d) UWn
- (e) -U [n|W|Wn]

where *n* is the number of instructions to execute before returning control to the console. You can abort the U command before *n* steps have been executed by typing **Ctrl+Break** at the console.

Format (e) differs from the analogous T command in that SID disables the display of intermediate pass points (while the pass count is greater than 1). In this case, only when the pass count reaches 1 is the pass information displayed (see the P command).

Examples

To trace without display 200H steps:

```
#u200
```

To trace without display 200H steps, suppressing the intermediate pass point display:

```
#u200
```


V (VALUE)

The V command displays the start and end addresses of the block of memory where the last file was loaded with the E or R commands. The V command has the format:

V

SID responds to the V command with a question mark if neither the R nor E commands have been used.

W (WRITE)

The W command writes the contents of a contiguous block of memory to disk. The following are the possible formats:

- (a) Wfilename
- (b) Wfilename,s,f

where *filename* is the filename and file type of the disk file to receive the data, and *s* and *f* are the 20-bit first and last addresses of the block to be written. If the segment is not specified in *f*, SID uses the same value that was used for *s*.

With format (a), SID assumes the *s* and *f* values from the last file read with an R command. If no file was read with an R command, SID responds with a question mark. This format is useful for writing out files after patches have been installed, assuming the overall length of the file is unchanged.

With format (b), where *s* and *f* are specified as 20-bit addresses, the low four bits of *s* are assumed to be 0. Thus the block being written must always start on a paragraph boundary.

If a file with the name specified in the W command already exists, SID deletes it before writing a new file.

Examples

To write to the file **test.exe** the contents of the memory block read into by the most recent R command:

```
#wtest.exe
```

To write the contents of the memory block 40:0 through 40:3FFF to the file **test.exe** on drive B::

```
#wb:test.exe,40:0,3fff
```

X (EXAMINE CPU STATE)

The X command allows you to examine and alter the CPU state of the program under test. The X command has the formats:

- (a) X
- (b) Xr
- (c) xf

where *r* is the name of a CPU register and *f* is the abbreviation of one CPU flag. Format (a) displays the CPU state as follows:

```
AX      BX      CX      .      .      .      SS      ES      IP
- - - - -      xxxx xxxx xxxx      .      .      xxxx xxxx xxxx
instruction                                     memory_value
```

The nine hyphens at the beginning of the line indicate the state of the nine CPU flags. Each position can be either a hyphen, indicating that the corresponding flag is not set (0), or a 1-character abbreviation of the flag name, indicating that the flag is set (1). Table 14-2 shows the abbreviations of the flag names.

Table 14-2 Flag Name Abbreviations

Character	Flag name
O	Overflow
D	Direction
I	Interrupt Enable
T	Trap
S	Sign
Z	Zero
A	Auxiliary Carry
P	Parity
C	Carry

The *instruction* is the disassembled instruction at the next location to be executed, which is indicated by the CS and IP registers.

If an instruction references memory, the contents of the referenced location(s) are displayed in the *memory_value* field, preceded by an equals sign. Either a byte, word, or double word value is shown, depending on the instruction. In addition to displaying the machine state, the first format changes the values of the default segments back to the CS and DS register values, and the default offset for the L command to the IP register value.

Format (b) allows you to alter the registers in the CPU state of the program being tested. The *r* following the X is the name of one 16-bit CPU register. SID responds by displaying the name of the register followed by its current value. If you type a carriage return, the value of the register is not changed. If you type a valid expression, the contents of the register are changed to the value of the expression. In either case, the next register is then displayed. This process continues until you enter a period or an invalid expression, or the last register is displayed.

Format (c) allows you to alter a flag in the CPU state of the program being tested. SID responds by displaying the name of the flag followed by its current state. If you type a carriage return, the state of the flag is not changed. If you type a valid value, the state of the flag is changed to that value. Only one flag can be examined or altered with each *Xf* command. Set or reset flags by entering a value of 1 or 0.

Examples

To change registers starting with BP:

```
#xbp
```

To change BP to hex 2B64:

```
BP=1000 2b64
```

To change SI to decimal 12345:

```
SI=2000 #12345
```

Z (Display the 8087 coprocessor registers)

This command displays the registers of an attached 8087 maths coprocessor if one is present.

14.3 Default segment values

SID has an internal mechanism that keeps track of the current segment value, making segment specification an optional part of a SID command. SID divides the command set into two types of command according to the segment a command defaults to if you do not specify a segment value in the command line.

The first type of command, pertaining to the code segment, includes the A (Assemble), L (List), P (Pass Points) and W (Write) commands. These commands use the internal type-1 segment value if no segment value is specified in the command.

When invoked, SID sets the type-1 segment value to the first available free segment.

- ❑ When an E command loads a file, SID sets the type-1 segment value to the value of the CS register.
- ❑ When an R command reads a file, SID sets the type-1 segment value to the base segment where the file was read.
- ❑ When an X command changes the value of the CS register, SID changes the type-1 segment value to the new value of the CS register.
- ❑ When SID regains control from a user program after a G, T, or U command, it sets the type-1 segment value to the value of the CS register.
- ❑ When an A or L command explicitly specifies a segment value, SID sets the type-1 segment value to the segment value specified.

The second type of command, pertaining to the data segment, includes the B (Block Compare), D (Display), F (Fill), M (Move), S (Set), and SR (Search) commands. These commands use the internal type-2 segment value if no segment value is specified in the command.

When invoked, SID sets the type-2 segment value to the first available free segment.

- ❑ When an E command loads a file, SID sets the type-2 segment value to the value of the DS register.
- ❑ When an R command reads a file, SID sets the type-2 segment value to the base segment where the file was read.
- ❑ When an X command changes the value of the DS register, SID changes the type-2 segment value to the new value of the DS register.
- ❑ When SID regains control from a user program after a G, T, or U command, it sets the type-2 segment value to the value of the DS register.
- ❑ When a B, D, F, M, S, or SR command explicitly specifies a segment value, SID sets the type-2 segment value to the segment value specified.
- ❑ When evaluating programs that have identical values in the CS and DS registers, all SID commands default to the same segment value unless explicitly overridden.

Table 14-3 summarizes SID's default segment values.

Note

The G (Go) command does not fall into either group because it defaults to the CS register.

Table 14-3 Summary of SID Default Segment Values

Command	Type-1	Type-2
A	*	
B		*
D		*
E	@	@
F		*
G	@	@
H		
I		
L	*	
M		*
P	%	
QI		
QO		
R	*	@
S		*
SR		*
T	@	@
U	@	@
V		
W	*	
X	@	@

Key to Table 14-3

- * Use this segment default if none specified; change default if specified explicitly.
- @ Change this segment default.
- % Use this segment default if none specified.

14.4 Assembly language syntax (A and L commands)

In general, the syntax of the assembly language statements in the A and L commands is standard 8086 assembly language. Several minor exceptions appear here.

- ❑ Up to three prefixes (LOCK, repeat, segment override) can appear in one statement, but they all must precede the opcode of the statement. Alternatively, a prefix can be entered on a line by itself.
- ❑ The distinction between byte and word string instructions is made as follows:

byte	word
LODSB	LODSW
STOSB	STOSW
SCASB	SCASW
MOVSb	MOVSW
CMPSB	CMPSW

- ❑ The mnemonics for near and far control transfer instructions are as follows:

short	normal	far
JMPS	JMP	JMPF
CALL	CALLF	
RET	RETF	

- ❑ If the operand of a CALLF or JMPF instruction is a 20-bit absolute address, it is entered in the form:

ssss:oooo

where *ssss* is the segment and *oooo* is the offset of the address.

- ❑ Operands that can refer to either a byte or word are ambiguous and must be preceded either by the prefix "BYTE" or "WORD". These prefixes can be abbreviated to "BY" and "WO". For example:

INC BYTE [BP]

NOT WORD [1234]

Failure to supply a prefix when needed results in an error message.

- ❑ Operands that address memory directly are enclosed in square brackets to distinguish them from immediate values. For example:

ADD AX,5; add 5 to register AX

ADD AX,[5]; add the contents of location 5 to AX

- ❑ The following are the forms of register indirect memory operands:

[*pointer_register*]

[*index_register*]

[*pointer_register* + *index_register*]

The *pointer_registers* are BX and BP, and the *index_registers* are SI and DI. Any of these forms can be preceded by a numeric offset. For example:

ADD BX, [BP+SI]

ADD BX, 3[BP+SI]

ADD BX, 1D47[BP+SI]

Part 3

Configuring

Running applications

YOU can run most applications simply by following the manufacturer's installation instructions. However, in some cases, it is possible to improve the performance of an application by changing some of the parameters in the application, in the operating system itself, or perhaps in both. This chapter explains what can be done.

The last section of this chapter gives a number of worked examples of possible batch files. These can be used to facilitate the running of applications in a secure, multiuser environment.

Caution This chapter invites you to run multiple copies of individual applications. However, you should only try to do this where the software license agreement permits it.

15.1 Overview

Applications written specifically for this operating system contain default information provided by the manufacturer to make them run efficiently in a multiuser, multitasking environment. Usually there is no need to adjust the operating system to run these **.cmd** applications. Simply follow the vendor's own installation instructions.

Most DOS applications have been developed to run by themselves in a single-user environment. Therefore, they may waste processor time searching for user input or use up too much memory, preventing other applications from running concurrently. You can overcome this problem by changing certain parameters in the operating system that affect the way applications run.

There are three sorts of setting which you can modify:

☐ System-wide settings

When you first install the operating system, system-wide defaults are automatically set up. These settings will be used every time you start or reset your computer and will apply to all applications running on the system.

☐ DOS session settings

You can override the system-wide defaults to give each DOS session its own parameter settings. These settings will only affect the applications running in that DOS session.

☐ Application settings

Applications can be assigned parameter settings individually. Information is entered into the application's program files using the Program Information Editor (PIFED) command. The operating system will use these parameters every time the application is loaded and the settings will apply to all users. The PIFED settings override both system-wide defaults and DOS session settings.

In many cases, these three categories are not mutually exclusive. Experiment to find the best way of resolving your particular configuration issues; bear in mind that the more local the modifications you make, the less risk you run of affecting other users on the system adversely.

15.2 System-wide settings

This section discusses the system-wide defaults that can be configured using the SETUP program or by altering statements in the **cconfig.sys** file.

Memory allocation

The amount of memory an application uses while running under the operating system is an important factor affecting its performance. (See Appendix D, “Memory management,” for a description of the different types of memory and an explanation of the terms used when discussing memory.)

The amount of memory (excluding LIM memory) that is available to applications in each DOS session is called the *Transient Program Area* or TPA. You can display the TPA size that an application will be allocated by entering the SHOW /M command in the DOS session in which you intend to run the application.

The amount of memory allocated to each application is defined by the `Maximum memory per process` option in the INSTALL and SETUP programs (see Chapter 1, “Doing an installation”). The default size is 656 Kbyte.

In a system with plenty of memory, allocation of memory will not be a problem. The maximum memory per process should be set to be equal to or greater than the largest TPA available. (The ways in which the TPA can be maximized are discussed in Chapter 16, “Customizing your system,” and Appendix D, “Memory management.”)

In systems where memory is limited, it may be necessary to restrict memory available to applications to below that of the largest possible TPA. This will prevent a few users from using up all the available memory and stopping other users from working.

LIM memory allocation

If you are running programs such as large databases and spreadsheets, you may have enabled LIM memory when you install the operating system. (These programs must have been

designed specifically to use the LIM memory specification.) Using LIM provides extra memory for the application's data; the amount of memory is set by the `Maximum LIM memory per process` option in the `INSTALL` and `SETUP` programs. The default size is 1024 Kbytes and this is in addition to the `Maximum memory per process` value discussed in the last section. More memory can be assigned if required by individual applications.

The most efficient way to use LIM is to use `SETUP` to set up the page frame in the TPA. Then use the `PIFED` command to set your applications up to use LIM.

When you are running more than one application which uses LIM memory on a system where memory is limited, set the LIM memory allocation to lower than 1024 Kbytes.

For more information about LIM see Chapter 16, "Customizing your system," and Appendix D, "Memory management." The `PIFED` options are described later in this chapter.

Idle detection

Interactive DOS programs can waste a lot of processor time checking for user input. By default, the operating system detects this *idle processing* and saves processor time for other applications.

Detection of idle processing can be enabled or disabled in **`cconfig.sys`** using the `IDLE` statement. The default setting is `IDLE=ON`.

Caution Take care when altering system-wide settings because they affect all users and all DOS sessions.

Using printers

In a multiuser environment we recommend that you use the print spooler `CSPOOL` to control printing. This queues data sent from different applications so that you can continue working even though the printer may have been busy at the time the data was sent to it.

Applications written for the operating system are aware of how to attach to and detach from the printer, making their use much easier.

However, most applications written for single-user DOS operating systems do not try to free the printer when they have finished printing and therefore keep control of the printer until the application is stopped. Using the spooler relieves this problem.

The printer used by a DOS session is defined with the PRINTMAP command.

Note that, if you do not use spooling, only one application at a time can send data to the printer.

The command interpreter

The operating system has its own command interpreter in addition to the outer level command environment which is used by most applications. This command interpreter is called **command.com**. **Command.com** provides a special environment in which terminate-and-stay-resident (TSR) programs and certain other programs can run without affecting other applications.

On machines that have another DOS operating system installed, **command.com** should be left in the default directory, **c:\osutils**. On machines with no other operating system installed, you are recommended to shift **command.com** to the root directory of the boot drive.

The COMSPEC variable defined in the SET command provides the path for the command interpreter; for example the default is:

```
SET COMSPEC=c:\osutils\command.com
```

If you move the **command.com** file to the root directory, change the COMSPEC setting accordingly.

The SET COMSPEC statement can be placed in **autoexec.bat** files. It can also be placed in the **cconfig.sys** file.

You can invoke the command interpreter by typing COMMAND at the command prompt.

15.3 DOS session settings

A more flexible way of improving performance is to enter operating system commands in the DOS session that the application will be run in. In this way, only that particular DOS session is affected.

You may find it useful to write a batch file of commands for each of your most frequently used applications and then run the relevant batch file just before you start an application. If you are unsure about how to create batch files see Chapter 10, "Batch processing."

The MEMSIZE setting

The MEMSIZE command limits the amount of memory available to an application in the DOS session that the command was issued from. In systems with limited memory, use this command to set a value lower than the maximum available TPA displayed by the SHOW /M command.

The LIMSIZE setting

To limit the maximum amount of LIM memory available in a DOS session, use the LIMSIZE command. If you are running several applications which use LIM memory, you may need to reduce the LIMSIZE from the system-wide value set by the Maximum LIM memory per process option.

The BANK setting

The BANK command defines how a program runs in memory. The default setting of ON allows the application to run in banked (paged) memory and is appropriate for most circumstances.

If you set BANK to OFF, you reduce the amount of memory available for other programs to be paged in and out of memory. Some programs may not be able to run and the performance of the others may be affected.

The IDLE setting

Idle detection can be altered for each DOS session using the **IDLE** command. The system-wide default is **IDLE=ON** and is appropriate for most circumstances. In those few cases in which the operating system might interfere with an application that only appears to be wasting processor time (for example in an application performing searching or sorting), you should set **IDLE=OFF**.

The SUSPEND setting

The **SUSPEND** command controls whether an application is suspended when it is switched into the background, that is, when you display another DOS session.

Using **SUSPEND = ON** allows greater system efficiency. It frees CPU resources and is more efficient than simply relying on idle detection to “freeze” applications in the background because of the overhead of the idle detection itself.

The NETDRIVE setting

Many applications designed to run on a DOS Local Area Network (LAN) can be run successfully. One computer with a main console and serial consoles can be made to function both as multiple workstations and as a LAN server.

To run properly, DOS LAN applications must recognize which physical drives are local and which are remote. They must also know when files are on the LAN server, since such files might be accessed there by several users at once, and so require file and record locking. The operating system provides this information through the **NETDRIVE** command.

NETDRIVE works on a per-DOS session basis. It can be set up using the **INSTALL** and **SETUP** programs, or you may want to put it in your **autoexec.bat** file or one of the **startnnn.bat** files. Alternatively, you may wish to put it in a batch file for each of the applications you use, as shown in the batch file examples later on in this chapter.

To set drive d: to local, use the format:

```
NETDRIVE d: /L
```

To set drive d: to remote or networked, use the format:

```
NETDRIVE d: /R
```

To display the current settings for a drive, use the format:

```
NETDRIVE d:
```

A useful construct in a batch file can be to save the existing NETDRIVE settings, for example:

```
NETDRIVE c: >oldnetdr.bat
```

then, having called the program you wish to run, execute the saved NETDRIVE settings:

```
CALL oldnetdr.bat
```

and finally remove the temporary batch file:

```
DEL oldnetdr.bat
```

15.4 Application settings

Using PIFED

You may find it more convenient to set parameters for an application individually using the Program Information Editor (PIFED). This will mean that every time you use the application it will start with the correct parameters *overriding* the system-wide defaults and DOS session settings.

Note

The PIFED settings will affect all users who run the application. Run a separate batch file if you want an application to use different parameter settings.

PIFED allows you to add or modify the program information embedded in a DOS file header. Once you have used PIFED on a program, the information is permanently stored in the header.

This program information is used by the operating system each time the program is loaded. For example, it is used to allocate the correct amount of memory for the program, to prevent

corruption of interrupt vectors and to stop other processes from accessing devices manipulated directly by a DOS process.

Because the information is stored with the program, it is used irrespective of the global settings in **cconfig.sys** and overrides any settings the user may have entered on the command line to control a particular DOS session.

There are two ways to use PIFED; entering the changes at the command line or by editing the PIFED screen for a program.

Using the PIFED screen

If you enter a program name without options, for example:

```
C>PIFED ws.exe
```

(including drive and path information if necessary) a message at the top of the screen will tell you whether program information has already been updated using PIFED, or whether the operating system is using default values. The default values will usually enable your file to run successfully, but you can often improve the performance of programs by changing the information settings.

If you choose to modify existing PIFED values, a second screen is displayed showing the current Program Information settings for that program. If PIFED has not been used on this program before, default values are shown.

The fields to complete depend on whether the file is a **.com** or **.exe**, or a **.cmd** file.

Fields for .com and .exe files

For **.com** and **.exe** files, the fields to complete are as follows:

Program name

This is a reminder of the name of the program whose program information you are altering.

Memory usage

The memory usage defaults will probably not need to be altered. However, you may find that increasing the “minimum memory required” enables the program to run more quickly. Similarly, if you know a program always allocates for itself more memory than it needs, you can reduce the “maximum

memory desired” and free more memory for other programs to use.

Screen usage

PIFED allows you to specify one of six different types of screen usage for your program. Refer to your program’s documentation to decide which type of screen usage best suits your program. You can toggle through the available options when the cursor is at the “screen usage” heading by either pressing the **Spacebar** or typing the option number. Types of screen usage are as follows:

- ☐ Program has no special requirements (24 or 25 lines).

Use one of these options for programs that only write simple messages to the screen. You can choose for the program to use 24 lines (option 0) or 25 lines (option 1). Choosing the 24-line option allows the status line to be displayed while the program is running.

- ☐ Program uses ANSI escape sequences (24 or 25 lines).

All programs that require the **ansi.sys** driver require at least this screen usage type, and may require the more complex usage types described below. (Note that **ansi.sys** is built in and does not need to be loaded specifically.)

- ☐ Program makes ROS calls (25 lines).

Select this option for programs that make calls to the ROM operating system.

- ☐ Program directly writes to screen, makes ROS calls and uses ANSI escape sequences.

This option, the default, is used for programs that write directly to screen. If you do not know the program’s requirements, use this default.

- ☐ Program uses graphics.

Select this option for programs that turn the screen to graphics mode.

Program uses LIM memory

Select **Yes** in this option if the program is written to use LIM memory. Otherwise select **No**.

If you select **Yes**, a 64 Kbyte page frame is used to page data. The location of this page frame depends on whether a page

frame was set up in upper memory by the INSTALL or SETUP programs. If it was, this page frame is used. If it was not, a page frame is set up in the TPA when the application is started. This is the more effective way to use memory, as the page frame only exists while the application using LIM is running. Other programs are not effected.

For more information about LIM, see Appendix D, “Memory management.”

Program runs only in the foreground

This allows you to specify explicitly that a program runs only in the foreground (i.e. when the DOS session it is running in is being displayed).

Note that this PIFED option overrides the current SUSPEND command setting while the application is running.

Program may run in banked memory

The default here is Yes. If you specify No, the operating system cannot page the program in and out of conventional memory.

Note that this PIFED option overrides the current BANK command setting while the application is running.

Program waits in idle loop

Some programs execute loops while they are idle and waiting for user input. This can be wasteful on computer processing time. The operating system has the ability to look for a program using *idle loops*, and reduce the slice of time that it allocates to that program. However, other programs execute idle loops while they are also doing something else (such as recalculating a spreadsheet), and you do not want the operating system to look for idle loops in this case. If you are sure the program only uses idle loops when waiting for further user input, set this option to Yes. If not, set it to No. (Note: This option overrides the current IDLE command setting while the application is running.)

Program requires aligned memory

If you select Yes, the program is only run if there is enough contiguous memory (memory in adjoining 16 Kbyte blocks). This is necessary for some device drivers which use DMA (hardware Direct Memory Access) to transfer data, and which might otherwise overwrite blocks of paged memory allocated to another program. For more information, see the description of the ALIGN command earlier in this chapter. (Note: This option

overrides the current **ALIGN** setting while the application is running.)

Program uses interrupt vectors

The default values for interrupt vectors enable most programs to run. Increasing the range of interrupt vectors may enable you to run certain complex programs that write directly to the interrupt vectors. Decreasing the range of interrupt vectors for a program that doesn't write directly to the interrupt vectors may marginally improve the program's performance.

Fields for .cmd files

There are only two fields to complete for .cmd files:

Program runs only in the foreground

This option is the same as that for .com and .exe files.

Program may run in banked memory

This option is the same as that for .com and .exe files.

Leaving the PIFED screen

When you have completed the PIFED screen for a file, press **Enter**. You can then choose to:

- ☐ Install the changes. These will now be the program defaults until PIFED is used on the program again.
- ☐ Continue using PIFED, by returning to the previous screen to continue editing with the values you just set.
- ☐ Revert to the original setting, by returning to the previous screen to continue editing with the *initial* values.
- ☐ To exit from PIFED without saving the changes.

Using PIFED at the command line

If you enter the PIFED options at the command line, the syntax is:

```
PIFED [filespec] [/B] [/I label=setting] [/R]
```

This is useful if you want to include the PIFED command in a batch file that you run to update several program files.

You can enter several labels on one line. The labels correspond to the fields in the PIFED screen and are shown by the on-line help displayed by typing:

A>PIFED /H

Setting up applications for terminals

PC Compatible Terminals

To run an application using a PC Compatible Terminal (for example, a Wyse® 60/120 set for PC terminal emulation or a PC running PCTERM), configure the application for one of the following:

- ☐ IBM® monochrome display adapter
- ☐ IBM ROM BIOS interface
- ☐ ANSI terminal
- ☐ VT52 terminal

The operating system converts an application's display output into the control sequences and characters required by any of these options.

Note

Options for graphic output should be disabled.

Some applications can detect the screen type automatically, and will work correctly both on the main screen and on the attached PC Terminals, whereas some do not.

If the software will not autodetect the display type then it will have to be configured for MDA (IBM Monochrome Display Adapter) to work on the attached PC Terminals. Many of this type of application will also work on the main screen when configured for MDA even if the display adaptor is an EGA or VGA. Not all do however, and it may be necessary to run MODE MONO on the main screen to persuade the application to work.

Some applications keep configuration information for each workstation and therefore can maintain correct information as long as the user always uses the same terminal type. For

example, if you are **accounts:a_user1** and always use the main screen, then the applications should be configured for the main screen. If however you log in at a PC Terminal then the applications may not be configured correctly and could fail. If you always use a PC Terminal, then it does not matter which PC Terminal is used.

Non-PC-compatible terminals

To run applications using a simple serial terminal, such as a Wyse 50, you must run the application's own configuration program to adapt it according to the type of serial terminal you are using.

Many DOS applications will not run on simple serial terminals because they require display and keyboard features which are not supported by these terminals. Other applications may require a special screen driver.

Using a mouse

Install applications that use a mouse as a Personal System/2® mouse and use the **mouse.sys** mouse driver supplied with the operating system. If the application does not support a Personal System/2 mouse, set it up to use a Microsoft® Bus mouse.

The mouse driver automatically adjusts to the physical mouse being used, that is the mouse defined to the application does not need to be the physical mouse type actually used to drive the application. The physical type of mouse connected to a port is defined using the INSTALL or SETUP program. See Chapters 1, "Doing an installation," and 16, "Customizing your system," for more information.

The **mouse.sys** driver is sufficient to drive most text based applications, but does not contain the full functionality required for some graphics applications. To obtain full mouse support load an appropriate third-party **mouse.com** driver at that DOS session, before running the application.

If the **mouse.sys** driver is already loaded then the third-party **mouse.com** driver will believe a Personal System/2 mouse is present and use the **mouse.sys** driver to provide the mouse position information. This has the advantage that if you switch DOS sessions then the mouse can continue to be used by the foreground DOS session.

If a third-party **mouse.com** driver is used without the **mouse.sys** driver then the mouse will be attached to that DOS session only and will not be available to the new foreground DOS session if you switch DOS sessions.

Remember that **mouse.com** will look at COM1 and COM2 to see if a mouse is attached, so if you are using a serial mouse then make sure that the logical COM1 or COM2 device points to the correct AUX port. The PRINTMAP command will do this.

For example, if AUX 0 is the real COM1, AUX 1 is the real COM2 or Multiport 1 and AUX 2 is the Multiport 2, then:

```
PRINTMAP COM1=AUXn  
PRINTMAP COM2=NUL
```

will allow **mouse.com** to find the mouse on AUX n , believing it is on COM1.

Configuring applications for printing

DOS programs should, where possible, be configured to use LPT1 as the printer device so that the operating system can intercept the data and route it to the correct printer. If the DOS program is configured to print to device PRN then this means the same as printing to LPT1.

The default printer for the main console and any attached serial terminals is the printer on LPT1, the first parallel port of the computer, known as PRINTER 0.

The PRINTMAP utility allows easy assignment from logical names to physical printer number so that any user can easily access any printer. By default the DOS device name LPT1 will refer to the printer on the parallel port of the main console (or the printer port on the back of the serial terminal).

If the printer is connected to either Serial Port 1 (COM1) or Serial Port 2 (COM2) on the main console, and the application is configured for LPT1 or PRN, then the PRINTMAP utility should be used to route the data from the application to the printer.

Only configure the application to print to COM1 or COM2 if it is impossible to use PRN or LPT1 routed to the serial port with PRINTMAP.

Multiple printers

Only one program can use the printer at a time. If there are multiple printers attached to the system, it may be necessary to use the PRINTMAP command to select the appropriate printer before running the program.

Data hand-shaking protocols

A computer may send data to a printer faster than it can print. Therefore the computer needs to know when to pause and when to send data to the printer again. This is done by *data hand-shaking*.

If a printer is attached to a parallel port on the computer, no configuration for data hand-shaking protocols is necessary, as it is done automatically.

When a printer is attached to a serial port you must configure the operating system for the correct baud rate and data hand-shaking protocol (see the SETUP program and the MODE command). Some applications can do their own protocol handling when configured to print to COM1 or COM2, but it is more efficient to use the operating system to provide this support.

Direct memory access (DMA)

Some block device drivers use Direct Memory Access of hardware (DMA) to transfer data to and from a device. Applications which are loaded from, or access data on drives supported by such a device driver, must be loaded in aligned memory to make logical memory coincide with physical memory in 64 Kbyte segments. This is done using the ALIGN statement in **cconfig.sys**. (See the Chapter 16, "Customizing your system," for an explanation of **cconfig.sys**.)

The system-wide default is ALIGN=OFF. To load all applications as aligned, set ALIGN=ON in **cconfig.sys** or set each application individually using PIFED. The PIFED setting overrides the **cconfig.sys**.

Most programs will not require ALIGN=ON. Possible exceptions are some copy protected programs and tape or disk drivers. If an incorrect DMA access is made, the operating system detects it, aborts the program and issues an error

message. If this happens, set `ALIGN=ON` for the application or driver using the `PIFED` command and reload the program.

Updating the `cconfig.sys` file

Cconfig.sys is the operating system's configuration file. Some applications automatically insert information into a **config.sys** file (the DOS equivalent) when they are installed on the hard disk. Where this occurs, make sure the relevant information is copied to the **cconfig.sys** file, or use `SETUP` to modify the configuration.

Remember that **cconfig.sys** entries affect all DOS sessions, so make sure the settings you copy are valid for all the applications on the system, not just the application most recently installed.

Note

If you need to alter the `FILES` statement in your **cconfig.sys** file, you must use `SETUP` to modify per-DOS-session and global `FILES` settings. Editing the **cconfig.sys** file directly will not have the desired result.

Terminate-and-stay-resident (TSR) applications

Terminate-and-stay-resident (TSR) applications, such as Sidekick®, will only stay resident when loaded from the operating system command interpreter **command.com**. **Command.com** is loaded automatically when you start a TSR and most TSRs will work correctly when loaded in this way.

If you experience problems, load **command.com** at the command line by entering `COMMAND` *before* you start the TSR. You may then use `EXIT` to terminate **command.com** when you have finished using the TSR application.

TSRs are only active in the DOS session in which they were first loaded.

Running multiple copies

Some DOS applications which have not been designed to run in a multiuser environment can have problems when running more than one copy. For example, they try to open several temporary files of the same, fixed, name. This can be overcome by creating a separate subdirectory for each user and installing a copy of the application in each subdirectory. Each user then runs the application from his or her own subdirectory.

Caution Only run multiple copies of applications when your software licence permits it.

Using a shared program subdirectory

Installing multiple copies has the disadvantage of taking up a lot of disk space. To avoid this, set up the program and operating environment to allow shared access of program and common files through a shared subdirectory.

The program and its overlays can usually be shared and loaded from a single subdirectory specified via the PATH and APPEND commands. The default directory should be the one used for work files and private data.

The program overlays should be set to read-only with the ATTRIB command so that APPEND can allow access to them by multiple users.

Preventing file conflict

Applications that share a subdirectory may cause file conflict if temporary work files are created in the shared subdirectory and are not given unique names. If this is likely to occur, configure the application so that work files are created in users' private subdirectories. If this is not possible, a separate copy of the application will have to be installed in each user's subdirectory.

Problems when running multiple copies

If you have any problems while trying to run a second copy of an application from a shared program subdirectory, it is possible that the file attributes on some of the program files are incorrect. Check by following this procedure:

- 1 Determine which files are created during the program installation.

These files can usually be identified by the date and time shown by the DIR /L command. These are the files that may need to be left as they are, allowing both read and write. They contain information related to the configuration of the program, and will change when any program parameters are altered.

The date and time will be that of the system date and time at installation, so ensure that they were correct before installation. Usually the program files and subsidiary parts such as program overlays and help message files will have the date and time that was on the original program disks. The date and time are only updated when the files are changed, or written to, and not when copied.

- 2 With the ATTRIB command, set files on which the date and time have not changed to read-only. This should include the files with extensions **.com**, **.exe** (and possibly the overlays **.ovr**, **.ovl** as well).
- 3 Run the application from one DOS session, and then when it is running switch to a second DOS session.
- 4 Use DIR to see if any new files have been created either in the private subdirectory or in the subdirectory from which the program is loaded (via PATH and APPEND). These could be the temporary work-files.
- 5 Now try running in two DOS sessions simultaneously. If any errors occur it is possible that they are caused by incorrect ATTRIB settings.

Try experimenting using the ATTRIB command.

Conflicts when loading overlays may occur if when attempting to start the second copy of the program from the hard disk, the program cannot find the overlay because it is already in use and not in read-only mode. The program then thinks that, because it cannot find the overlay, the user is running on a diskette-based machine and issues one of the following error messages:

```
Insert System Disk 2 and press ENTER or press
Ctrl-C to abort.
```

```
Disk Not Ready
```

```
Sharing Conflict
```

In these circumstances, try to identify which file is causing the problem and set it to read-only.

Copy-protected applications

Some software copy-protection schemes depend on timing loops which make disks unreadable when run with fast processors. Most computer manufacturers provide a "slow down" feature to allow diskette access at the standard speed. Consult your manufacturer's documentation if you have trouble running such programs.

If you have problems with copy-protection, try setting `ALIGN=ON` as many copy-protection schemes use direct access to the diskette controller. (See also "Direct Memory Access" earlier in this section.)

There may be some difficulty in installing copy-protected applications when upper memory is enabled.

15.5 Worked examples

This section contains a number of worked examples of batch files. These batch files simplify running applications in a secure multiuser environment.

For a sample batch file that produces a menu of commonly-used applications and commands, see Appendix G, "Sample menuing batch file."

Preparing the system

In these examples it is assumed that the user has installed the operating system into the **c:\osutils** subdirectory, and that all the third-party general purpose utility programs are in **c:\tools**.

It is assumed that these sample applications will be installed for use by all user groups. For example, you could have an Accounts group that wished to access the spreadsheet programs and a Sales group that wished to access the database programs. With the simplest secure system, the files associated with an application are owned by the Superuser who has full access permissions to them, but are set to read-only for the "world." This means that, in this example, both Accounts and Sales can use all the applications.

To restrict access so that only one group can access an application, the files associated with the application must be donated to that group with the XATTRIB command using the /A:username option, where *username* is one of the users within the group.

- 1 Log in to the operating system as Superuser so that you have full access permissions to the whole system.
- 2 Make group **accounts** with members **a_user1**, **a_user2**, **a_user3** and so on.

(Because user names are unique it is sensible to reflect the group name in the user name, for example **a_user1** in **accounts** or **s_user1** in **sales**.)

- 3 Make sure that a common application subdirectory has been defined and that all users have sufficient access rights to the directory, for example:

```
XATTRIB c:\apps /D /O:RWD /W:RWD /G:RWD /A:superuser
```

- 4 Make sure that a common batch-file subdirectory has been defined and that all users have sufficient access rights to the directory, for example:

```
XATTRIB c:\osutils\batch /D /O:RWD /G:RWD /W:RWD /A:superuser
```

- 5 All files in this directory should be set to read-only permission for normal users, for example:

```
XATTRIB c:\osutils\batch\*.*/O:RWD /G:RWD /W:R /A:superuser
```

(Remember to do this each time you add new batch files into this directory.)

- 6 Determine the system-wide disk drive letters that are free for use, to be defined as the user's home drive, for example H:. Create a **global.bat** file in **c:\osutils\batch** containing the SUBST command to point to each user's unique home directory:

```
@ECHO OFF
REM global.bat in c:\osutils\batch
SUSPEND OFF
PROMPT $p$g
SUBST h: /D >NUL
SUBST h: . >NUL
H:
PATH h:\;c:\osutils;c:\osutils\batch;c:\tools
```

- 7 Create an initial **autoexec.bat** suitable for all users for most applications. Keep it in the **c:\osutils\batch** directory and donate a copy to each user in each user's home directory. It can be modified by each user, but the CALL global.bat line should not be removed:

```
@ECHO OFF
REM autoexec.bat sample should be placed
REM in each user's home directory
CALL c:\osutils\batch\global.bat %1 %2 %3 %4 %5
REM change or add lines after this if a user
REM needs any changes to their copy of this file
```

- 8 The following batch file lines give the sample **autoexec.bat** file to **a_user1** in group accounts:

```
COPY c:\osutils\batch\autoexec.bat c:\accounts\a_user1
XATTRIB c:\accounts\a_user1\autoexec.bat /A:a_user1
```

- 9 Ensure that the global number of file handles and the maximum for each application are sufficient. Use SETUP to modify these values (do not edit **cconfig.sys** directly).

Paradox® v3.5

The following example illustrates a multiuser installation of Paradox v3.5:

- 1 Login as Superuser.
- 2 Make the **c:\apps\pdox35** and **c:\apps\pdoxdata** subdirectories:
 MD c:\apps
 MD c:\apps\pdox35
 MD c:\apps\pdoxdata
- 3 Install Paradox as instructed in the Paradox documentation, with the programs being installed to **c:\apps\pdox35** and the Network Control File to **c:\apps\pdoxdata**. Select Network Type as "6 Other."
- 4 Ensure that all users have sufficient rights to read from **pdox35** and read/write/delete within **pdoxdata**:
 XATTRIB c:\apps\pdox35 /S /O:RWD /G:/R /W:R /A:superuser
 XATTRIB c:\apps\pdoxdata /O:RWD /G:RWD /W:RWD /A:superuser
- 5 Provide freer access permissions to the sample data, so that all users can access them:
 XATTRIB c:\apps\pdox35\sample /O:RWD /G:RWD /W:RWD /A:superuser
 XATTRIB c:\apps\pdox35\sampapp /O:RWD /G:RWD /W:RWD /A:superuser

Alternatively, if the data is only to be changed from one user group then the data files can be donated to a user within that group, setting "owner" and "group" rights to RWD.

For example, to nominate **s_user1** within **sales** as owner of the files, type:

 XATTRIB c:\apps\pdox35\sample /O:RWD /G:RWD /W:/A:s_user1
 XATTRIB c:\apps\pdox35\sampapp /O:RWD /G:RWD /W:/A:s_user1
- 6 Temporarily define h: as **c:\apps\pdoxdata**:
 SUBST h: /D
 SUBST h: c:\apps\pdoxdata
- 7 Run the Paradox Custom Script:
 C:
 CD c:\apps\pdox35
 PARADOX custom

- 8** Set the Paradox directories up for all users, so that the Private Directory is **h:** and the Working Directory is **c:\apps\pdxdata** (shared by all users).
- 9** Select the network option when saving the **paradox.cfg** file, and save to **h:**.
- 10** When back at the command prompt, set correct access permissions for the **paradox.cfg** and **.net** files as follows:


```
XATTRIB c:\apps\pdxdata\paradox.ctg /O:RWD /G:RWD /W:RWD /A:superuser  
XATTRIB c:\apps\pdxdata\paradox.net /O:RWD /G:RWD /W:RWD /A:superuser
```
- 11** Copy the Paradox configuration file (**paradox.cfg**) to each user's home directory and donate it to the user, for example:


```
COPY h:\paradox.cfg c:\accounts\a_user1  
XATTRIB c:\accounts\a_user1\paradox.cfg /A:a_user1  
  
COPY h:\paradox.cfg c:\sales\s_user1  
XATTRIB c:\sales\s_user1\paradox.cfg /A:s_user1
```
- 12** The following example batch file should be built and put in the **c:\osutils\batch** subdirectory:

```
@ECHO OFF  
REM paradox.bat to run Paradox v3.5  
PATH >oldpath.bat  
PATH c:\apps\pdx35;%path%  
PARADOX.EXE %1 %2 %3 %4 %5 -share -user %username%  
CALL oldpath.bat  
DEL oldpath.bat
```

Note: to run Paradox in multiuser mode it is not necessary to have **NETDRIVE c: /r** because **PARADOX -share** will have the same effect.

The user can now run Paradox v3.5 by invoking the batch file **paradox.bat** in his or her home directory, **h:**.

Microsoft® Word v5.0

The following example illustrates a multiuser installation of Microsoft Word v5.0:

- 1 Login as Superuser.
- 2 Install Word 5 for a network server using Word 5 SETUP.EXE, installing to **c:\apps\word5**, installing video support for the main console adapter (eg VGA) and for the PC Terminals (MDA - Monochrome Display Adaptor) and selecting all the printer drivers that are relevant.
- 3 Make a batch file **word.bat** in **c:\osutils\batch** containing:

```
@ECHO OFF
REM word.bat to run Microsoft Word v5.0
PATH >oldpath.bat
PATH c:\apps\word5;%path%
SET MSWNET=h:\
NETDRIVE c: /R
WORD.EXE %1 %2 %3 %4 %5
NETDRIVE c: /L
SET MSWNET=
CALL oldpath.bat
DEL oldpath.bat
```

- 4 Set access permissions for the users:
`XATTRIB c:\apps\word5 /S /O:RWD /G:R /W:R /A:superuser`
- 5 Login as a user within the group (and repeat this paragraph for each user within the group) and, while in home directory **h:**, issue the command
`c:\apps\word5\setup user`. Follow the normal Word 5 installation for a workstation, configuring to subdirectory **h:**, selecting the screen and printers for that user. Do not select any mouse configuration, and when leaving SETUP do not request Word 5 to modify **config.sys** or **autoexec.bat**.

The user can then run Word 5 by invoking the batch file **word.bat** in his or her home directory, **h:**.

Lotus 1-2-3® Release 2.2 Server Edition

The following example illustrates a multiuser installation of Lotus 1-2-3 Release 2.2 Server Edition:

- 1 Log in as Superuser.
- 2 Make the following subdirectories:
MD c:\lotshare
MD c:\lotshare\123.v22 (The Lotus license directory)
MD c:\apps\123r22 (The 1-2-3 programs directory)
MD c:\apps\123r22\allways (The Allways directory)
- 3 Install Lotus 1-2-3 Release 2.2 to the above directories, as instructed in the LAN administrator's guide.
- 4 Set access permissions with

```
XATTRIB c:\apps\123r22 /S /O:RWD /G:R /W:R /A:superuser  
XATTRIB c:\lotshare /D O:RWD /G:RWD /W:RWD /A:superuser  
XATTRIB c:\lotshare\123.v22 /O:RWD /G:RWD /W:RWD /A:superuser
```

Note: A symptom of incorrect access permissions (for example where you are unable to read, write and delete) to **c:\lotshare** or **c:\lotshare\123.v22** is that an unexpected error message may occur when starting Lotus 1-2-3 Release 2.2 saying that the Lotus 1-2-3 license count has been exceeded.

- 5 With "world" access to **c:\apps\123r22** set to read-only, a member of group **accounts** will only be able to read the sample spreadsheet files in read-only mode. These files will only be able to be opened without a Lotus 1-2-3 file reservation. It is recommended that you copy the data files to a directory such as **c:\apps\123r22\data** and allow fuller access to these files either with:

```
XATTRIB c:\apps\123r22\data\*. * /O:RWD /G:RWD /W:RWD /A:superuser  
or
```

```
XATTRIB c:\apps\123r22\data\*. * /O:RWD /G:RWD /W: /A:a_user1  
where a_user1 is a member of group accounts.
```

- 6 To use the Lotus 1-2-3 File-Admin-Reservation-Get and Release commands it is necessary to ensure that `NETDRIVE C: /R` has been performed in the batch file which invokes Lotus 1-2-3.
-

- 7** Create the **123r22.bat** file in the **c:\osutils\batch** directory with the following contents:

```
@ECHO OFF
REM 123r22.bat to run LOTUS 1-2-3 Rel. 2.2
PATH >oldpath.bat
PATH c:\apps\123r22;%path%
NETDRIVE c: /R
123.EXE %1 %2 %3 %4 %5
NETDRIVE c: /L
CALL oldpath.bat
DEL oldpath.bat
```

- 8** Create the **lotus.bat** file in the **c:\osutils\batch** directory with the following contents:

```
@ECHO OFF
REM lotus.bat to run LOTUS.COM from Lotus 1-2-3 Rel. 2.2
PATH >oldpath.bat
PATH c:\apps\123r22;%path%
NETDRIVE c: /R
LOTUS.COM %1 %2 %3 %4 %5
NETDRIVE c: /L
CALL oldpath.bat
DEL oldpath.bat
```

- 9** Login as each user in turn (where the user's home directory is **h:**) and go into **c:\apps\123r22** and run **newuser.bat** with the command `newuser h:`, then while still in **c:\apps\123r22** run the **INSTALL** program to perform a normal first-time install to **h:\123** to select the appropriate screen and printer drivers. (Select the Universal Text Display for text screen, and no graphics if you will be running on a serial PC-Terminal.) Repeat for each user.

- 10** To install **ALLWAYS** it is necessary to go into **c:\apps\123r22\allways** and run the **AWSETUP.EXE** program, configuring to drive **h:**. During **AWSETUP** select IBM monochrome display adapter if you will be running Lotus 1-2-3 on a serial PC terminal. Repeat this while logged in as each user.

To run Lotus 1-2-3 Release 2.2, a user should invoke either **lotus.bat** or **123r22.bat** from his or her home directory, **h:**.

DataEase® v4.2 LAN

The following example illustrates a multiuser installation of DataEase v4.2 LAN:

- 1** Log in as Superuser.
- 2** Make the DataEase directory:
MD c:\apps\dease
- 3** Install DataEase v4.2 using INSTALL.EXE on DataEase disk 1.

During the DataEase install procedure select DataEase (640K version) and the network type Banyan VINES®.
- 4** Create the **dease.bat** file in **c:\osutils\batch** that will be used to invoke DataEase:

```
@ECHO OFF
REM dease.bat to run DataEase 4.2 LAN
PATH >oldpath.bat
PATH c:\apps\dease;%path%
SET dename=%username%
NETDRIVE c: /R
DEASE.EXE %1 %2 %3 %4 %5
NETDRIVE c: /L
SET dename=
CALL oldpath.bat
DEL oldpath.bat
```

- 5** Set correct access rights for the users with:
XATTRIB c:\apps\dease /S /O:RWD /G:RWD /W:RWD /A:superuser

Note that the DataEase program requires read-write-delete permission in the program directory.
- 6** Any drive on which shared data files to be used by DataEase exist must be set as a networked drive with the NETDRIVE command (for example NETDRIVE c: /R), so that DataEase will know that it should do file and record locking.

The user can run DataEase by invoking the batch file **dease.bat** in his or her home directory, **h:**.

Quattro Pro® v2.0

The following example illustrates a multiuser installation of Quattro Pro v2.0:

- 1 Log in as Superuser.
- 2 Create the following subdirectories:
 MD c:\apps\qpro
 MD c:\apps\qprodata
- 3 Install Quattro Pro v2.0 into **c:\apps\qpro** using INSTALL.EXE on Quattro Pro disk 1.
- 4 Run QPUPDATE to define serial numbers, to select network type 6, "Other," and to change the location of the **qpro.net** file to **c:\apps\qprodata**.
- 5 Set access permissions so that each user can read and write to the **qprodata** directory:
 XATTRIB c:\apps\qprodata /O:RWD /G:RWD /W:RWD /A:superuser
 The private directory for each user will be **h:** (which is defined to point to the user's home directory).
- 6 For each user create a **.mp** file for the user's menu preference. For the example user **a_user1** in group **accounts** the commands required are:
 CD c:\apps\qpro
 C:
 MPMAKE xxx.mu c:\accounts\a_user1\ (where xxx is quattro,123, or q1)
 XATTRIB c:\accounts\a_user1 /O:RWD /G:/W:/A:a_user1
- 7 Create a default file (**.rf**) in each user's private directory. For the example user **a_user1** in group **accounts** the commands required are:
 COPY c:\apps\qpro\rsc.rf c:\accounts\a_user1
 XATTRIB c:\accounts\a_user1\rsc.rf /O:RWD /G:/W:/A:a_user1
- 8 Repeat the steps for **.mp** and **.rf** files for each user, then delete the **rsc.rf** file from the **c:\apps\qpro** subdirectory.
- 9 Make the **c:\apps\qpro** subdirectory read-only for the users:
 XATTRIB c:\apps\qpro /O:RWD /G:R /W:R /A:superuser

- 10** Create the **qpro.bat** file in **c:\osutils\batch** which will be used by the users to invoke Quattro Pro.

```
@ECHO OFF
REM qpro.bat to run Borland Quattro Pro v2.0
PATH >oldpath.bat
PATH %path%;c:\apps\qprodata;c:\apps\qpro
NETDRIVE /R c:
Q.EXE %1 %2 %3 %4 %5
NETDRIVE /L c:
CALL oldpath.bat
DEL oldpath.bat
```

The path to the private directory, here **h:**, must be before the path to the **qpro** subdirectory. In this example **%path%** will pick up **h:\;c:\osutils;c:\osutils\batch;c:\tools** as defined in the file **global.bat**.

The Quattro Pro documentation says that the program **SHARE** must be used , if Paradox files are to be accessed from within Quattro Pro. The **SHARE** program functionality is built in to this operating system, and so it is not necessary to use the **SHARE** command.

- 11** Any drive on which shared data files to be used by Quattro Pro exist must be set as a networked drive with the **NETDRIVE** command (for example **NETDRIVE d: /R**), so that Quattro Pro will know that it should do file and record locking.

The user can run Quattro Pro by invoking the batch file **qpro.bat** in his or her home directory, **h:**.

Lotus Manuscript® v2.1 LAN

The following example illustrates a multiuser installation of Manuscript v2.1 LAN:

- 1** Log in as Superuser.
- 2** Create a temporary drive pointing to **c:\apps**, for example:
SUBST i: c:\apps
- 3** Install Lotus Manuscript v2.1 as instructed in the LAN administrators guide. The install procedure creates a directory **c:\apps\ms** and installs the application into it.
- 4** Create the following license subdirectories:
MD c:\apps\ms\lotshare
MD c:\apps\ms\lotshare\mscript.v21
- 5** Go into **c:\apps\ms** and run the Manuscript COUNT program. This should find a valid license directory on drive L: (the special "load drive" floating drive). Set the license count according to the terms of your Lotus Manuscript license agreement.
- 6** Set access permissions for the users:
XATTRIB c:\apps\ms /O:RWD /G:R /W:R /A:superuser
XATTRIB c:\lotshare /D O:RWD /G:RWD /W:RWD /A:superuser
XATTRIB c:\lotshare\mscript.v21 /O:RWD /G:RWD /W:RWD /A:superuser
- 7** Create the **ms.bat** file in the **c:\osutils\batch** directory with the following contents:
@ECHO OFF
REM ms.bat to run LOTUS Manuscript v2.1
PATH >oldpath.bat
PATH c:\apps\ms;%path%
NETDRIVE c: /R
MS.EXE %1 %2 %3 %4 %5
NETDRIVE c: /L
CALL oldpath.bat
- 8** Log in as each user in turn (where the user's home directory is **h:**). Go into **c:\apps\ms** and run **newuser.bat** with the command:
NEWUSER h:

To run Lotus Manuscript v2.1, the user should invoke the batch file **ms.bat** from his or her home directory, **h:**.

WordPerfect® v5.1

The following example illustrates a multiuser installation of WordPerfect v5.1:

- 1** Log in as Superuser.
- 2** Install WordPerfect 5.1 using the WordPerfect install program to the directory **c:\apps\wp51**.
During the install select network type "0 Other."
Towards the end of the install select **c:\apps\wp51** as the subdirectory for the system setup files.
- 3** Set the correct access rights:
`XATTRIB c:\apps\wp51 /S /O:RWD /G:RWD /W:R /A:superuser`
- 4** Create the **wp.bat** file in **c:\osutils\batch** containing:

```
@ECHO OFF
REM wp.bat to run WordPerfect v5.1
PATH >oldpath.bat
PATH c:\apps\wp51;%path%
NETDRIVE c: /R
WP.EXE %1 %2 %3 %4 %5
NETDRIVE c: /L
CALL oldpath.bat
DEL oldpath.bat
```
- 5** Run the WordPerfect NWPSETUP program to create the correct environment for each user.

To run WordPerfect, a user should invoke the batch file **wp.bat** from within his or her home directory, **h:**.

LetterPerfect® v1.0

The following example illustrates a multiuser installation of LetterPerfect v1.0:

- 1** Log in as Superuser.
- 2** Install LetterPerfect 1.0 using the LetterPerfect install program to the directory **c:\apps\lp10**.

During the install select network type "0 Other."

Towards the end of the install select **c:\apps\lp10** as the subdirectory for the system setup files.

- 3** Set the correct access rights:

`XATTRIB c:\apps\lp10 /S /O:RWD /G:RWD /W:R /A:superuser`

Create the **lp.bat** file in **c:\osutils\batch** containing:

```
@ECHO OFF
REM global.bat to run LetterPerfect v1.0
PATH >oldpath.bat
PATH c:\apps\lp10;%path%
NETDRIVE c: /R
LP.EXE %1 %2 %3 %4 %5
NETDRIVE c: /L
CALL oldpath.bat
DEL oldpath.bat
```

- 4** Run the LetterPerfect NLPSETUP program to create the correct environment for each user.

To run LetterPerfect, a user should invoke the batch file **lp.bat** from within his or her home directory, **h:**.

Q&A® v3.0 LAN

The following example illustrates a multiuser installation of Q&A 3.0 LAN:

- 1** Log in as Superuser.
- 2** Install Q&A 3.0 LAN using the normal Q&A installation procedure to the directory **c:\apps\q&a**.

- 3** Set the correct access rights:

```
XATTRIB c:\apps\Q&A /S /O:RWD /G:RWD /W:R /A:superuser
```

- 4** Create the **q&a.bat** file in **c:\osutils\batch** containing:

```
@ECHO OFF
REM q&a.bat to run Q&A v3.0
PATH >oldpath.bat
PATH c:\apps\q&a;%path%
APPEND c:\apps\q&a /X
NETDRIVE c: /R
QA.COM %1 %2 %3 %4 %5
APPEND;
NETDRIVE c: /L
CALL oldpath.bat
DEL oldpath.bat
```

Note: using this batch file will lose any previous settings of APPEND.

To run Q&A, a user should invoke the batch file **q&a.bat** from within his or her home directory, **h:**.

SuperCalc® 5

The following example illustrates a multiuser installation of Supercalc 5:

- 1 Log in as Superuser.
- 2 Make drive c: appear to be a network drive with the command `NETDRIVE c: /R`.
- 3 Create the directory `c:\apps\sc5`:
`MD c:\apps\sc5`
- 4 Create the directory `c:\apps\sc5\keyfile`:
`MD c:\apps\sc5\keyfile`
- 5 Set the correct access rights:
`XATTRIB c:\apps\sc5 /S /O:RWD /G:R /W:R /A:superuser`
- 6 Install SuperCalc 5 using the SuperCalc ACTIVATE program to the directory `c:\apps\sc5`, using the network path `\\netdrive\c\apps\sc5`.
- 7 When prompted for the read or write keyfile access paths, use the network path `\\netdrive\c\apps\sc5\keyfile`.
- 8 Pledge the appropriate number of copies of SuperCalc.
- 9 Create the `sc5.bat` file in `c:\osutils\batch` containing:

```
@ECHO OFF
REM sc5.bat to run SuperCalc 5
PATH >oldpath.bat
PATH c:\apps\sc5;%path%
NETDRIVE c: /R
SC5.COM %1 %2 %3 %4 %5
NETDRIVE c: /L
CALL oldpath.bat
DEL oldpath.bat
```

To run SuperCalc 5, a user should invoke the batch file `sc5.bat` from within his or her home directory, `h:\`.

Note

The operating system supports the network drive mappings that are required by SuperCalc and ACTIVATE, but does not support the mapping of network printers via SuperCalc or ACTIVATE. Printers should be mapped using the normal `PRINTMAP` command before going into SuperCalc.

Framework® III v1.1

The following example illustrates a multiuser installation of Framework III version 1.1:

- 1** Log in as Superuser.
- 2** Create a directory for the application, for example:
MD c:\apps\fw3
- 3** Set the drive to remote:
NETDRIVE c: /R
- 4** Install Framework III to the directory you have created, using the FWSETUP installation program.
- 5** Install the Framework mail package to the **c:\apps** directory, using MHSINST on the Framework Mail Setup disk.
- 6** Create the following **fwmail.bat** file in **c:\osutils\batch**, to run FWMAIL and so complete configuration of the mail package:

```
@ECHO OFF
REM fwmail.bat to run Framework III FWMAIL program
PATH >oldpath.bat
PATH c:\apps\fw3\mhs\exe;c:\apps\fw3;%path%
REM save old netdrive setting
NETDRIVE c: >oldnetdr.bat
NETDRIVE c: /R
SET MV=c:\apps\fw3\
SET CLR=0
FWMAIL.EXE %1 %2 %3 %4 %5
SET fwpersonal=
SET clr=
SET mv=
REM restore old netdrive setting
CALL oldnetdr.bat
CALL oldpath.bat
DEL oldnetdr.bat
DEL oldpath.bat
```

- 7** Create the **fw3.bat** file in **c:\osutils\batch** that will be used to invoke Framework III:

```
@ECHO OFF
REM fw3.bat to run Framework III v1.1
PATH >oldpath.bat
```

```
PATH c:\apps\fw3;c:\apps\fw3\mhs\exe;%path%
REM save old netdrive setting
NETDRIVE c: >oldnetdr.bat
NETDRIVE c: /R
SET fwpersonal=h:\
SET mv=c:\apps\fw3\
SET clr=0
FW.COM %1 %2 %3 %4 %5
SET fwpersonal=
SET clr=
SET mv
REM restore old netdrive setting
CALL oldnetdr.bat
CALL oldpath.bat
DEL oldnetdr.bat
DEL oldpath.bat
```

- 8** Set the correct access rights for the users with:
XATTRIB c:\apps\fw3 /S /O:RWD /G:R /W:R /A:superuser
XATTRIB c:\fwnetctl.300 /S /O:RWD /G:RW /W:RW /A:superuser
- 9** Set the Framework default directory to **h:**, so that each user has a separate default directory.
- 10** Set to remote any drive that contains shared data files that will be used by Framework III, for example:
NETDRIVE c: /R

To run Framework III, a user should invoke the batch file **fw3.bat** from within his or her home directory, **h:**.

dBASE IV® v1.1

The following example illustrates a multiuser installation of dBASE IV version 1.1:

- 1 Login as Superuser.
- 2 Install dBASE IV v1.1, using the "Full" install option of the dBASE install program, and install to **c:\apps\dbase4**.
- 3 Run DBSETUP while in **c:\apps\dbase4**, set the dBASE IV default directory to **h:** so that each user has a separate default directory, and save **config.db** into **c:\apps\dbase4**.
- 4 Copy **config.db** into the home directory of each user, and set the correct file attributes, for example:

```
COPY c:\apps\dbase4\config.db c:\accounts\a_user1
XATTRIB c:\accounts\a_user1\config.db /A:a_user1
```
- 5 Create the following **dbase4.bat** batch file in **c:\osutils\batch**, which will invoke dBASE IV:

```
@ECHO OFF
REM dbase4.bat to run dBASE IV
PATH >oldpath.bat
PATH c:\apps\dbase4;%path%
NETDRIVE c: /R
DBASE.COM %1 %2 %3 %4 %5
NETDRIVE c: /L
CALL oldpath.bat
DEL oldpath.bat
```

- 6 Set to remote any drive that contains shared data files that will be used by dBASE IV, for example:
NETDRIVE c: /R

To run dBASE IV, a user should invoke the batch file **dbase4.bat** from within his or her home directory, **h:**.

Customizing your system

YOU may want to adapt your system to improve overall efficiency. There are several approaches, depending on what you want to achieve:

- ❑ You can make adjustments to most of the standard settings and defaults set during installation by using a program called **SETUP**.

Like **INSTALL**, **SETUP** has menus of options that are easy to select and change. **SETUP** takes your changes and automatically updates your system files and the configuration files **cconfig.sys** and **autoexec.bat** for you. These changes apply globally the next time you start up your computer.

Using **SETUP** is described in section 16.1 and Chapter 1, “Doing an installation.”

- ❑ You can edit the configuration files **cconfig.sys** and **autoexec.bat** directly, using **EDITOR**, if you prefer. Note that some advanced features can only be set up by directly editing the configuration files. **Cconfig.sys** and **autoexec.bat** are described in this chapter.
 - ❑ You can add installable device drivers to your **cconfig.sys** file.
 - ❑ You can use a number of commands to determine how resources are shared.
-

16.1 Using SETUP

To start SETUP, type:

```
C>SETUP
```

A welcome screen appears with general information about the SETUP program. When you press **Enter** to continue, you are asked for the name of the system file to be updated with the changes you are about to make. The default is the file from which the system currently running has been loaded.

The next screen asks whether you want to accept the current configuration and write it to the system file entered on the previous screen, or configure the system.

If you choose the latter option, a new screen is displayed with a number of options. If you only need to update one area of your system, for example memory, you do not need to work through all the screens.

SETUP screens are similar to those that appear when you install the operating system, and are described in the “Configuring the system” section of Chapter 1, “Doing an installation.”

SETUP has full context-sensitive help, just press the **F1** key on a highlighted option or when the cursor is positioned on the field you are interested in.

You can leave SETUP at any time and return to the system prompt by pressing **F10** and selecting the option **Exit** to the operating system. This will not change your existing system configuration.

When you have finished making changes, select the option **Write setup information to system file**. A new screen shows that the changes have been made.

You can go back at any time and rerun SETUP to change any values you have just set.

SETUP and system configuration files

The SETUP program changes the settings in two files that are specifically used to customize your system to exactly the way you want it. These files are **cconfig.sys** and **autoexec.bat**. Every time the operating system is started, it searches its root directory for these files and carries out the commands contained in them.

Cconfig.sys has its own set of commands, listed in section 16.2.

When you install the operating system a **cconfig.sys** file is built up and copied into the root directory of your hard disk.

To see how your **cconfig.sys** file is set up, enter:

```
C>TYPE cconfig.sys
```

You see something like this:

```
device=c:\osutils\mouse.sys /s
device=c:\osutils\vdisk.sys 1024 128 64
device=c:\osutils\cache.exe /s=768
install=c:\osutils\keyb.com uk
break=on
buffers=20
fastopen=512
lastdrive=e
country=001
```

The display will vary according to your individual system setup.

SETUP with an input file

INSTALL and SETUP automatically create an ASCII file that holds the system configuration values. This *system input file* is given the same name as the system file name you specify, but with a **.inp** extension.

Normally the input file is not used because the system file also contains the system configuration values. However, it can be used as an input file to SETUP to avoid working through the configuration screens. For example, if you know that one configuration worked well, you could save it (by renaming the input file) before you tried any more changes to the configuration. Then if they were unsuccessful, you could restore the old configuration using the SETUP command as follows:

```
SETUP /I systemfilename [inputfilename]
```

The system configuration values in the system input file will then be used instead of those in the system file. If you do not specify the *inputfilename*, the default (*systemfilename* with the extension **.inp**) is used.

You can edit the input file with an ASCII editor. See Appendix E, "Using a setup input file," for more on the subject.

16.2 Cconfig.sys commands

Although INSTALL and SETUP create certain **cconfig.sys** entries automatically, a number of entries can only be entered directly using an editor.

The **cconfig.sys** commands are explained on the following pages. They can be entered in any order. Examples given under each command show how to set these parameters directly in the **cconfig.sys** file.

Table 16-1 lists the commands that can be placed in **cconfig.sys**.

Table 16-1 Cconfig.sys Commands

Command	Explanation
?	Executes commands conditionally.
:label	Defines a label that can be referenced by a GOTO subcommand.
ALIGN	Determines whether programs are loaded in aligned (contiguous) memory.
BREAK	Allows you to set the default state of the BREAK command which lets you stop a program while it is running.
BUFFERS	Sets the number of disk buffers allocated.
COUNTRY	Selects the appropriate date and time format and currency symbol for your country.
DEVICE	Enables extra device drivers to be installed.
ECHO	Allows the display of messages on screen during startup.
ENVSIZE	Modifies the number of bytes allocated to the initial environment.
FASTOPEN	Increases the speed with which files are accessed when the same files are repeatedly used.
FILES	A dummy utility for DOS compatibility.
FLAGS	Increases the number of system flags.
GOTO	Used to jump to a label elsewhere in the file.
IDLE	Determines if idle detection will be used.
INSTALL	Allows you to load a program during startup.
LASTDRIVE	Sets the last valid drive letter.
NETWORK	Loads DR Net.
QUEUES	Increases queue buffer size.
REM or ;	During batch file execution displays the comment given on the same line as REM.
SET	Sets the default environment on a secure system.

?

Format

? ["remark"] command

Explanation

If you insert a ? at the beginning of a statement in your **cconfig.sys** file, when the commands in **cconfig.sys** are executed you are asked if you want this particular command carried out. This is most useful when applied to the **DEVICE=** statement so you can choose whether to install specific device drivers. For example the following statement in your **cconfig.sys** file:

```
?DEVICE = c:\osutils\vdisk.sys
```

means you are queried each time you reboot your system as to whether you want the device driver **vdisk.sys** loaded, like this:

```
DEVICE = c:\osutils\vdisk.sys (Y/N) ?
```

Type **Y** to load **vdisk.sys**, **N** to ignore it.

Alternatively, you can place a message string between the question mark and the **cconfig.sys** statement, enclosed in double quotation marks, for example:

```
?"Do you want vdisk.sys?"DEVICE = c:\osutils\vdisk.sys
```

then at startup you are prompted with your message:

```
Do you want vdisk.sys?
```

Answer by typing **Y** for Yes or **N** for No. This can be convenient if you have access to a network, as you can set up a prompt that asks you at startup whether you want to use the network.

Remarks

You can use ? with any of your **cconfig.sys** file statements.

The maximum length allowed for a command statement including a question mark and message string is 128 characters. Any statements longer than this are ignored.

:label

Format

:label

Explanation

:label defines a label that can be referenced by a GOTO **cconfig.sys** subcommand.

Only the first eight characters after the colon are considered part of the label by GOTO.

ALIGN

Format

ALIGN = ON|OFF

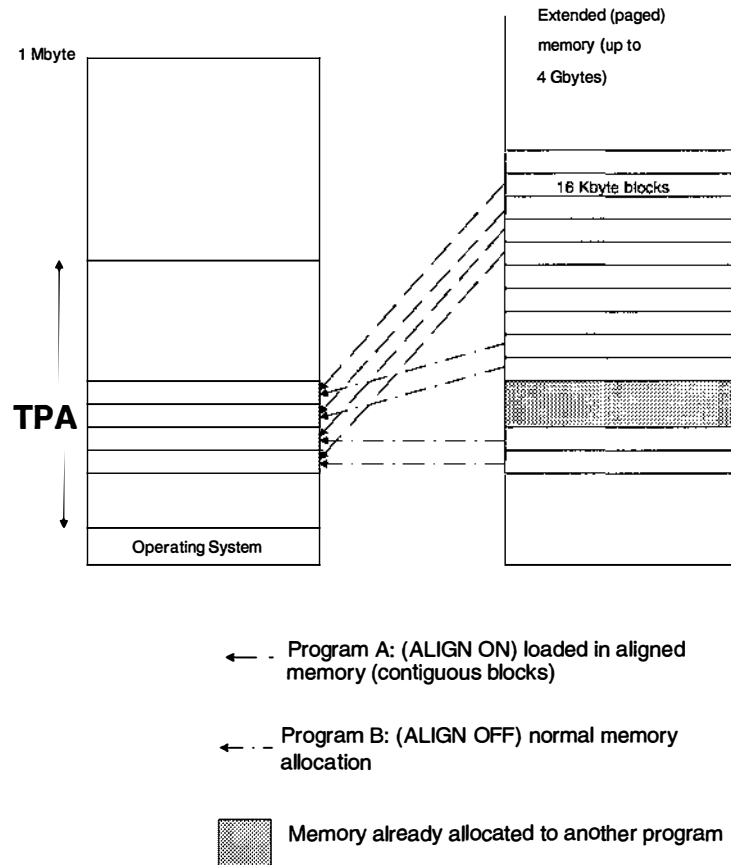
Explanation

There are some block device drivers that use DMA (hardware Direct Memory Access) to transfer data to and from the device. Any program that is loaded from, or that accesses data on drives supported by a device driver of this type must be loaded in aligned (contiguous) memory. The ALIGN statement in **cconfig.sys** causes the operating system to ensure all memory allocations are aligned.

Memory is paged in and out of the application processing memory (TPA) in 16 Kbyte blocks. However, a program that uses say 64 Kbytes is not necessarily allocated four adjoining 16 Kbyte blocks of expanded memory. Normally, this does not matter because the operating system is aware of the mappings and controls memory access. However, block device drivers that use DMA bypass the operating system: because they only provide a starting address when they write to memory, these drivers would overwrite the next four 16 Kbyte blocks of expanded memory, interfering with other programs.

Setting ALIGN = ON ensures that the operating system loads an application only if there is enough contiguous memory (adjoining blocks) as shown on the next page. This is a less-optimized memory allocation scheme but avoids memory being overwritten. Use this setting only if the device driver specifically requires it. Typical examples are tape streamer drivers and the installation of copy protected programs.

By default, the operating system does not guarantee to load programs in aligned memory (ALIGN=OFF), thereby optimizing memory allocations.

**Figure 16-1 Aligned Memory Allocation**

BREAK

Format

BREAK = ON|OFF

Explanation

You can stop a program when it is running by holding down the **Ctrl** key and pressing **Break** or **C**.

If BREAK is OFF (the default), the key sequence **Ctrl+Break** or **Ctrl+C** will only stop the program when it next reads from the keyboard, writes to the screen or writes to a printer.

When BREAK is ON, the operating system also looks for **Ctrl+Break** or **Ctrl+C** when the program is reading from or writing to a disk.

Remarks

BREAK = ON lets you quickly stop programs that do not use standard input and output operations very often, such as programs that perform lengthy calculations before printing the final output.

The BREAK command in the **cconfig.sys** file is overruled *for that particular DOS session* when you type a BREAK command at the command line. Equally, both are overruled for a particular application by the setting in a PIFED command.

Note

Many applications disable this feature so that you can only use the method documented by the application to stop it.

BUFFERS

Format

```
BUFFERS = nn
```

Explanation

Buffers are small areas (or blocks) of memory that the operating system uses during disk read or write operations. Buffers store the information that is being read or written. You can specify between 3 and 99 buffers.

Remarks

The number of buffers you specify generally affects the speed at which a program accesses files. The more buffers there are, the faster the system accesses files. However, each buffer occupies memory that would otherwise be used for the program itself and for your data. If you specify a large number of buffers, your program may actually run slower than if you specify a small number.

Your application documentation should suggest how many buffers you need. Between 10 and 40 is a sensible number; a few more may be needed if the program uses a large number of files. A desktop publishing program may need between 20 and 30, for example.

Example

The following entry in your **cconfig.sys** file sets the number of buffers to 20.

```
BUFFERS = 20
```

COUNTRY

Format

COUNTRY = *nnn*

Explanation

The country code (*nnn*) provides the date and time format and currency symbol appropriate for your country. For European countries, for example, the preferred date format is day/month/year, whereas for North America it is month/day/year and for Japan, year/month/day. In the United Kingdom the currency symbol is £ rather than \$ as used in the United States.

The default value may already be set for your own country and may not need to be changed. Table 16-2 shows the country codes available.

The country code does not include information about the keyboard type; that information can be provided using the `INSTALL = keyb.com` command which is described later in this chapter.

Table 16-2 Country Codes

Country	Code (<i>nnn</i>)
Australia	061
Austria	043
Belgium	032
Canada (English)	002
Canada (French)	002
Denmark	045
Finland	358
France	033
Germany (West)	049
Israel	972
Italy	039
Japan	081
Latin America	003
Middle East	785
Netherlands	031
Norway	047
Portugal	351
Spain	034
Sweden	046
Swiss	041
United Kingdom	044
United States	001

Example

This entry in your **cconfig.sys** file converts international currency symbols, date and time to Italian format.

```
COUNTRY = 039
```

DEVICE

Format

DEVICE = filespec [options]

Explanation

The **DEVICE** command enables device drivers to be installed. Device drivers for the keyboard, screen, disk drives, printer and auxiliary devices are automatically loaded as part of the operating system startup sequence.

Note

Ansi.sys is built into the system. Refer to Appendix F, “Escape code sequences,” for information about the supported escape sequences.

Whenever you add a new piece of computer hardware to your system, like a printer or mouse, you should be supplied with a device driver if one is required. Device drivers are programs that are loaded alongside the operating system to operate hardware devices that are not recognized as a default part of your system. Device drivers usually have the extension **.sys**. You need to copy the supplied device driver onto the disk from which you start the system. Then use the **DEVICE** command in **cconfig.sys** to ensure the driver is found and loaded each time the system is started.

Three device drivers are supplied with the operating system and can be loaded with the **DEVICE** command:

cache.exe

mouse.sys

vdisk.sys

These drivers are described in section 16.3.

Remarks

If you precede a device statement with a question mark before the device driver is loaded, you are prompted as to whether you want it installed. This gives you the option every time you start the system of omitting a particular device driver.

Most device drivers are hardware specific, but some are designed for software functions, like **vdisk.sys**, which is used to set up a memory disk on your system.

Example

The following statement installs the device driver **vdisk.sys**:

```
DEVICE = c:\osutils\vdisk.sys
```

ECHO

Format

ECHO message

Explanation

ECHO allows the display of messages on screen during startup.

It is useful for including questions and explanatory messages as part of your **cconfig.sys** file, and these messages are displayed at startup in the order they are inserted in the **cconfig.sys** file.

Example

The following command puts a message on the screen indicating that a mouse driver is being loaded:

```
ECHO Loading mouse driver
```

ENVSIZE

Format

ENVSIZE = nnn

Explanation

This command modifies the number of bytes allocated to the initial environment. The minimum number is 128 and the maximum is 32768.

Use ENVSIZE with applications that require a larger than usual environment.

The ENVSIZE command affects all DOS sessions, and an alternative is to run a copy of **command.com** with the required environment size from a particular DOS session. Only that DOS session is affected. Type:

```
C>COMMAND.COM /E:size
```

Example

To set the environment size to 640 bytes, enter:

```
ENVSIZE = 640
```

FASTOPEN

Format

FASTOPEN = nnnnn

Explanation

FASTOPEN reduces the time taken to access files on a hard disk which are being accessed repeatedly. It does this by remembering the locations of the files on disk, so that the operating system does not have to navigate along the path every time the file is opened.

The operating system sets up a special table of these filenames in memory and you can set the size of this table by specifying the number of entries.

The minimum number is 128 and the maximum is 32768.

Remarks

This command improves disk performance at the cost of memory space. Each table entry uses 2 bytes of memory. Having a larger table than you actually need will not increase your system's performance.

Note

If upper memory is available, the directory index table is stored there.

Example

This command sets up a table size of 512 entries:

FASTOPEN = 512

FILES

Format

`FILES = nnn`

Explanation

This command is only supported for DOS compatibility.

To alter the number of files that can be opened by a program at the same time, invoke **SETUP** and modify the `Maximum open files per DOS session` field.

Remarks

You should only increase the default if your application documentation instructs you to, or if the program you are using returns error messages that the number of open files is insufficient.

FLAGS

Format

FLAGS = nnn

Explanation

FLAGS allows you to change the default number of system flags available. This number can be used by system-aware applications.

The maximum number of flags is 255.

Example

The following command sets the number of flags at 100:

FLAGS = 100

GOTO

Format

GOTO label

Explanation

GOTO is used to change the order in which commands are executed. It is often used together with a ? statement, so that if a condition is satisfied execution jumps to another part of **cconfig.sys**. The *label* parameter must identify a label that is present in the **cconfig.sys** file; GOTO can then direct the operating system to jump to the label.

Example

In the following sequence, if the user types Y for Yes, the operating system switches to the command after the label, and loads **vdisk.sys**:

```
? "Do you want a memory disk? (Y/N)" GOTO memdisk
.
.
.
:memdisk
DEVICE=vdisk.sys
```


IDLE

Format

IDLE = ON|OFF

Explanation

Many interactive DOS programs are written without consideration for the effect of running on a multiuser system. These programs look repeatedly to see if a key has been pressed. This wastes processor time that could be used to improve the performance of other users and programs running on the system.

When IDLE is set ON, the operating system detects that an application is idle and allows other programs to use the saved processor time until the application needs to process such events as a mouse movement or keyboard entry.

The IDLE statement in **cconfig.sys** is used to enable or disable idle detection globally. By default, idle detection is enabled (IDLE=ON).

Note

It is more usual to enable and disable idle detection for each application using the PIFED command. For example, a word processor in the middle of a long search or replace operation might appear to be idle but in fact be working. Such applications should have IDLE=OFF.

INSTALL

Format

```
INSTALL = filespec [options]
```

Explanation

The **INSTALL** command allows you to load a standard **.com** or **.exe** program during startup. You install the keyboard driver **keyb.com** in this way. Equally, the program could be a Terminate Stay Resident (TSR) program required by all users.

WARNING Standard DOS TSRs installed in this way are not multiuser-aware, and so may not operate successfully.

Options indicate the options to pass to the program given in the filespec.

Example

The following **INSTALL** statement in a **cconfig.sys** file loads the **keyb.com** program during **cconfig.sys** processing:

```
BUFFERS = 10  
INSTALL = c:\osutils\keyb.com IT
```

LASTDRIVE

Format

```
LASTDRIVE = drive_letter
```

Explanation

The operating system uses letters to identify particular drives. It automatically assigns a drive letter to each actual drive that it finds during the startup procedure (including any virtual disk drives). It then sets the last drive as being the drive letter assigned to the last drive it finds. All the remaining drive letters then become invalid.

The LASTDRIVE command enables you to set a different last drive, so that the operating system will recognize non-assigned drive letters.

Example

Suppose your computer has a diskette drive (drive A:), and a hard disk drive (drive C:). You have added two virtual disks (drives D: and E:) and you have a program that requires you to use the SUBST command with drives F: and G:. To use LASTDRIVE, type LASTDRIVE followed by the highest drive letter you will need:

```
LASTDRIVE = c
```

This means that the operating system recognizes drives F: and G: as well as drives A: through E:.

NETWORK

Format

```
NETWORK = filespec [/A]
```

Explanation

NETWORK loads the DR Net local area network at run time.

The /A switch produces a prompt asking you whether you wish to load DR Net.

Note

This command, if present, must occupy the first line in the **cconfig.sys** file.

Example

The following example produces a prompt asking whether you want to load DR Net, and if the answer is affirmative loads the file **netsys.smc**:

```
NETWORK = netsys.smc /A
```

QUEUES

Format

`QUEUES = nqcb, qb`

Explanation

`QUEUES` allows you to set the number of queues available to operating system-aware applications that use internal system queues. It also lets you establish the queue buffer space.

For `nqcb` enter a decimal value for the number of queue control blocks required. Queues created by transient programs and those Resident System Processes (RSPs) outside the 64K of the SYSDAT segment address each require a queue control block.

For `qb` enter a decimal value for the number of paragraphs of buffer space required. The amount of buffer area required for each queue is the message length times the number of messages.

The values you enter are absolute values and replace the default configuration values.

REM

Format

```
REM | ; [remark]
```

Explanation

REM allows a remark to be inserted into a **cconfig.sys** file.

REM can be replaced by a semicolon (;).

Use **REM** with a remark to clarify what a command does; use it on its own to add spacing for readability.

Example

To clarify what an entry does:

```
REM * Load the vdisk.sys driver  
DEVICE = vdisk.sys
```

SET

Format

```
SET [name=[parameter]]
```

Explanation

Use the SET command within **cconfig.sys** to set the default environment in a “secure” system with user accounts established.

The string (beginning with *name*) is inserted into a part of memory reserved for environment strings. Lowercase letters in the name are converted to uppercase when added to the environment. If the name was in the environment, it is assigned the new parameter.

If you type SET with no options you see a list of the strings already in the environment.

To delete a name from the environment, type SET with a name and an equals sign but no parameter.

16.3 Installable device drivers

This section describes the additional device drivers supplied with the operating system; they are summarized in Table 16-3. These drivers are automatically set up during installation by DEVICE and INSTALL. Equally they can be set up at a later date using the SETUP program.

Table 16-3 Installable Device Drivers

Device driver	Explanation
cache.exe	Allows you to set up an area of RAM memory to act as a buffer, storing a copy of data frequently read and written from disk, to speed up disk access.
keyb.com	Tells the system what national keyboard you are using.
mouse.sys	Provides multitasking mouse capabilities.
vdisk.sys	Sets aside a portion of RAM memory to act as if it was a very fast disk.

Note

The driver **ansi.sys** is built in; it does not need to be installed separately. (**Ansi.sys** provides extra options for the keyboard and screen. The escape code sequences supported are shown in Appendix F, “Escape code sequences”.)

CACHE.EXE

This driver enables you to set up an area of extended memory to act as a buffer, storing a copy of data that is frequently read from disk. This area of memory is called a *cache*.

Every time your applications attempt to access information on disk, a check is first made to see if a copy of the data is already in the cache. If it is, the copy is used instead, which is much faster than accessing the disk itself.

Caches can be very useful if your work involves applications that need to access the same data over and over, like a database or accounts program.

You can install **cache.exe** using the SETUP program or during installation. Alternatively, place a statement as follows in your **cconfig.sys** file:

```
DEVICE = c:\osutils\cache.exe /S=nnnn
```

The *nnnn* value is the buffer size in Kbytes (in the form /S=2048). This must be specified.

Remarks

If you want to fine tune your cache, start with the largest possible memory size, and then after running a typical application (that is one of the applications you use often), note the *use* and *hit rate* figures by displaying the cache status report (you can do this by typing **CACHE** at the command prompt). Note that a cache *hit* is when the cache is accessed and valid data is found within. Repeat this process, reducing the memory allocated to the cache each time and then noting the new use and hit values. Continue until the memory size is reduced to its lowest value before the hit rate begins to drop. This will vary according to the disk and applications used.

Examples

To install **cache.exe** as a device driver using 1 Mbyte of extended memory for caching, include the following command line in **cconfig.sys**:

```
DEVICE = c:\osutils\cache.exe /S=1024
```

KEYB.COM

Keyb.com provides the information needed to use national keyboards. The `INSTALL = keyb.comxx` command is automatically placed in **cconfig.sys** by **INSTALL** and **SETUP** if you choose a national keyboard other than the default US keyboard. Each national keyboard is represented by a code (*xx*) shown in Table 16-4.

Table 16-4 Keyboard Codes

Country	Code (<i>xx</i>)
Belgium	BE
Canada (French)	CF
Denmark	DK
Finland	SU
France	FR
Germany	GR
Italy	IT
Latin America	LA
Netherlands	NL
Norway	NO
Portugal	PO
Spain	SP
Sweden	SV
Swiss (French)	SF
Swiss (German)	SG
United Kingdom	UK
United States	US

Only one copy of **keyb.com** is loaded to provide national keyboard support for all terminals in the system.

MOUSE.SYS

Mouse.sys is a driver that provides the ability to use a PS/2 mouse or serial mouse with the operating system.

To load the driver, the following command needs to be in **cconfig.sys**:

```
DEVICE = c:\osutils\mouse.sys
```

This is done automatically by INSTALL and SETUP when you select to use a mouse either on the main console or on any terminal.

In INSTALL and SETUP define main console mouse type as being either a PS/2 or serial mouse, others are always serial. The mouse driver adapts automatically to the type of mouse being used.

Use with multiple applications

Mouse.sys provides the capability to run more than one DOS session and to use the same physical mouse with each one. At any time, the mouse controls the DOS session in the foreground, that is, the one you are displaying.

Installing applications to use the mouse

For information about using applications that have a mouse interface, see Chapter 15, "Running applications."

VDISK.SYS

Vdisk.sys lets you set aside a specified portion of extended memory to act as if it were a fast disk. This area of memory is called a memory disk.

A memory disk has a very fast access time, but it is only in effect while your computer is switched on. When you switch off or reset your computer, everything stored in a virtual disk is lost. This means that anything stored on it that you want to keep must be copied to a hard disk or diskette before switching off.

Vdisk.sys can be useful for storing temporary files used by some programs, or by programs that need to access a lot of information quickly and often, like a database or spreadsheet.

One virtual disk created by INSTALL or SETUP

Using INSTALL or SETUP, you can create one virtual disk, and a line is inserted into **cconfig.sys** to do this. You can create up to four virtual disks; simply copy the `DEVICE =` statement changing the option if necessary.

The full command is:

```
DEVICE = c:\osutils\vdisk.sys [disksize] [sectsize]  
[nnn]
```

WARNING Remember that any files stored on virtual disks are lost when the computer is switched off or when the system is rebooted. Any information you need to keep must be stored on diskette or on your hard disk.

Option	Meaning
<i>disksize</i>	This option specifies the size of the virtual disk in Kbytes. The default is 64 Kbytes, and the range 2 through 8000 Kbytes.
<i>sectsize</i>	Specifies the smallest amount of space reserved on the disk. The default is 128 bytes, alternative values are 256 or 512.
<i>nnn</i>	Specifies the maximum number of files to be written to the root directory of the virtual disk and can be between 2 and 512; the default is 64. The number governs how much virtual disk space is reserved for the root directory. Specifying a large number wastes space unless you plan to copy large numbers of files into the root directory of the virtual disk.

Example

The following command leads to a virtual disk of 200 Kbyte:

```
DEVICE = c:\osutils\vdisk.sys 200
```

Other supported drivers

Support for the following multiport devices is “built in” to the operating system:

- ☐ AST.
- ☐ Hostess.
- ☐ Arnet card.
- ☐ IBM Dual Asynch (COM3 and COM4).

The following generic drivers are supported:

- ☐ Other Multiport cards including the Arnet Smartport card.
- ☐ Tape streamer drivers.
- ☐ Drivers for CD ROM (only **.com** and **.exe** programs can access the CD ROM drive, not **.cmd** programs.)

These need to be loaded either by adding a **DEVICE = statement** in **cconfig.sys** or by using the “generic” option in **SETUP** and entering the driver name.

Example

To load the Arnet Smartport card, place the following command in **cconfig.sys**:

```
DEVICE = arnet.sys
```

16.4 The autoexec.bat file

The **autoexec.bat** file is a batch file that gets run each time your computer is started. **Autoexec.bat** sets up your computer the way you want it. One is created for you when you install the operating system. **Autoexec.bat** must be stored in the root directory of the disk from which you load the operating system (usually the hard disk).

If you have security enabled, in addition to this main **autoexec.bat** file each user can have a local **autoexec.bat** file in the home directory. The file is executed each time the user logs in, setting up parameters for that user only.

This section describes some of the commands that can be put in an **autoexec.bat** file, or that you may find already set up in the main **autoexec.bat** file. The commands mentioned are also discussed in Chapter 11, "Commands."

Note

The SETUP and INSTALL programs place two special labels in the **autoexec.bat** file, :OSBEG and :OSEND. SETUP then processes anything between the two labels. Put any additional statements for the system between these labels. Any other statements, for example for other operating systems, should be placed outside of the labels.

NETDRIVE

By default, INSTALL and SETUP define the system drive to be a remote drive. This allows correct multiuser access.

PATH and APPEND

The PATH command tells the operating system where to look for command, program, and batch files. APPEND tells it where to search for any other type of file.

You may want to put PATH and APPEND commands in the **autoexec.bat** file, tailored to your own directory hierarchy.

PROMPT

You can put a **PROMPT** command into your **autoexec.bat** file. **PROMPT** allows you to change the default system prompt. For example you could replace the familiar prompt with:

```
HAVE A NICE DAY!
```

or (more usefully) show the current directory, such as:

```
C:\WILLIAM\WUNDAWRD\REPORTS>
```

You can change the prompt whenever you like, but again you may prefer to do this automatically each time the system is started, by using the **autoexec.bat** file.

SET

You can put the **SET** command into your **autoexec.bat** file. **SET** is a command that helps programs to carry out their work by inserting a string of information into the command processor's environment. This string is then available to all programs and applications.

SHARE

The **SHARE** statement is provided for applications that specifically require it. The operating system, being multiuser, automatically allows file sharing.

SUSPEND

This command is set by **INSTALL** and **SETUP**. If it is set to **ON**, it suspends applications when they are in the background.

Station and DOS session numbers as parameters

If you want some commands to be executed only for some terminals or DOS sessions, you can pass **autoexec.bat** station and DOS session numbers as parameters. For example:

```
IF "%1"=="1" CSPOOL START
```

only loads **CSPOOL** on the first DOS session of the main console.

Three parameters can be used:

- ❑ Parameter %1 is the console number. This is a unique number for each DOS session on the system. The main console DOS sessions are numbers 1 upwards.
- ❑ Parameter %2 is the station number (00 for the main console, 01 and upwards for the other terminals on the system. The station number can be identified by using `SHOW /U.`)
- ❑ Parameter %3 is the DOS session number (1 up to a maximum of 8) on the station.

Starting applications from autoexec.bat

Another way the **autoexec.bat** file can be useful is to automatically load your favourite software. Suppose you have a word processor called WundaWord which is run by typing `ww` from a directory called **\wundawrd**. Normally you would need to type two commands to start WundaWord:

```
C>CHDIR \wundawrd
```

and then:

```
C>ww
```

You can put both these commands in the **autoexec.bat** file, so that WundaWord is loaded each time you start the computer.

For example you could add:

```
CD \wundawrd
WW
CD \
```

where:

Entry	Meaning
CD \wundawrd	Makes \wundawrd the current directory.
WW	Starts WundaWord.
CD \	Returns you to the root directory when you leave WundaWord.

16.5 START files

As well as the **autoexec.bat** file, the operating system supports START files.

START files are batch files that allow you to specify a different set of initial commands for a particular terminal or DOS session. These are then run on that terminal or DOS session instead of the commands in **autoexec.bat**. **Autoexec.bat** is run for each DOS session for which there is no START file. If you want to run the commands in **autoexec.bat** as well as those in the command file, call **autoexec.bat** from within the START file.

There can be one START file for each DOS session, identified in the name as:

`STARTxxxy.BAT`

where *xx* represents the station number (00 for the main console, 01 and upwards for the other terminals), *y* represents the DOS session number on the station (1 up to a maximum of 8). The station number is shown using `SHOW /U`.

For more information about creating and running batch files see Chapter 10, "Batch processing."

16.6 Sharing resources

All users share all the system resources whether "internal," such as memory and cache, or connected on serial cables such as printers, plotters and modems.

Printers, plotters and modems can only be accessed by one user at any time; other users must wait until the resource is free. There is one exception: if you control printing with a spooler.

Printing

Any user can print directly to a printer using the port assignments made with the `PRINTMAP` command. However, only one job can be printed on any one printer at any time. If other users try to print, their applications hang, waiting for the

printer to become free. They cannot carry on working until their job is printed. Many DOS applications do not specifically detach from the printer, and therefore even when the first job has finished printing, other applications could not use the printer until the first application has been stopped.

The operating system provides a print spooler, CSPOOL, to overcome these restrictions. CSPOOL “owns” all the ports unless they are specifically excluded from being a printer port. When a user asks to print, CSPOOL creates a print job of whatever needs to be printed. As soon as the job is created the user can carry on working.

The job is placed in the spool queue. When it reaches the top of the queue, it is printed.

When setting up a spool file, the spooler recognizes the end of a job by looking for one of three events:

- ☐ The application does not send any characters to the printer for a certain time, known as a time-out period.
- ☐ The application specifically sends a command to detach from the printer.
- ☐ The application is stopped.

PRINTMAP

The operating system assigns numbers to all printers (PRN n) and serial ports (AUX n). However, most DOS applications access printers by using the device names LPT1, LPT2 and LPT3, and serial ports by using the device names COM1 and COM2.

The PRINTMAP command allows you to decide which printer or serial port will be accessed when the application writes to a particular device name.

PRINTMAP assignments are made for each DOS session and only apply until the operating system is restarted. If you want the assignments to apply permanently, put the PRINTMAP statements in your **autoexec.bat** file. (For more information about PRINTMAP, see Chapter 13, “Printing.”)

CSPPOOL

CSPPOOL commands are described in Chapter 13, “Printing.” A summary of all the CSPPOOL commands is given in Table 16-5.

Table 16-5 CSPPOOL Commands

Command	Description
START	Starts the spooler.
QUIT	Removes the spooler from the system.
STATUS	Displays a list of jobs with their status.
STOP	Deletes a print job.
PRIORITY	Allows the priority of a queued job to be changed. Jobs are printed in order of priority; those with the same priority are printed in the order in which they were submitted.

Configuring the spooler

The **config.cpl** file contains information needed to run the print spooler. You may need to edit the information to configure the print spooler for your particular environment.

See Chapter 13, “Printing,” for more information.

Sharing other resources

This section discusses the other resources the users can share:

- ☐ Ports: the PRINTMAP and MODE (baud rate) commands.
- ☐ Memory.
- ☐ Disk space, access (NETDRIVE) and usage (file ownership).

Sharing ports

Each DOS session can be assigned an auxiliary port using the PRINTMAP command. More than one DOS session can use the same port. In this way users can share communications ports (for example, modems) and direct connections to printers that

are not using spooling. At any time, only one program has access to each port.

Most DOS applications access printers by using device names LPT1, LPT2, LPT3, and access serial ports by using device names COM1 and COM2. However, the operating system treats these devices names as logical names that can access any physical port, and uses the terms PRN*n* for printers and AUX*n* for serial ports. The mapping of logical names to physical ports is done using the PRINTMAP command. PRINTMAP can be entered on the command line in any DOS session to change the assignments for that DOS session (until the system is restarted). Enter the command without parameters to see the current assignments in a DOS session. In this way, one user can be using an application that outputs to LPT1 to print on a laser printer, while someone else uses the same application to print on a draft printer. For more information about PRINTMAP, see Chapter 13, "Printing."

One form of the MODE command is used to configure communications ports. See Chapter 13, "Printing," for more information. Note that when ports such as COM1 are defined in the MODE command, these are logical ports and are mapped to physical ports using the assignments defined with PRINTMAP.

Sharing memory

As long as there is enough memory available, users can start a program in each DOS session on their terminal. However, when all the available memory has been used up, an error message is displayed and the requested program is not be able to run until some memory is freed.

It is therefore important to ensure that each DOS session is not allocated more memory than necessary. This can be achieved in several ways:

- ❑ The most effective is to allocate only as much memory as a program requires using the PIFED command. Note that PIFED only controls the amount of conventional memory, not LIM memory. PIFED is described in Chapter 15, "Running applications."
 - ❑ The memory available for a DOS session can be controlled by using the MEMSIZE and LIMSIZE commands from within that DOS session. Each user should be encouraged to use these commands when they are running programs for which they know the memory requirements.
-
-

- ❑ The default memory allocations are determined by the **INSTALL** and **SETUP** options **Maximum memory per process** and **Maximum LIM memory per process**. These are global settings that apply if **MEMSIZE** and **LIMSIZE** are not used, or if a program without a **PIFED** header is run.

Disk and file sharing

All users can access any of the drives on the system and, unless system security is enabled or passwords assigned, every user has access to all the files on all these directories.

Access to files can be restricted if system security is in use; files and directories then have an owner (usually the creator) who controls access permissions. For more information about file ownership, access permissions and passwords see Chapter 8, “Protecting your information.”

The amount of disk space available to each user is unrestricted. It is therefore advisable to take regular backups so that files that are no longer used can be removed from the hard disk(s).

NETDRIVE allows multiuser DOS applications to behave as if they were being run on a networked or remote drive. Using **NETDRIVE**, drives can be designated as local or remote drives. For more information, see Chapter 11, “Commands.”

|

Error messages and troubleshooting



This appendix is divided into two sections:

- ☐ **Error messages.** An explanation of the operating system error messages.
- ☐ **Troubleshooting.** What to do if you have problems.

A.1 Error messages

This section lists operating system error messages in alphabetical order, giving the origin of the message followed by an explanation of it and where possible, suggestions for correcting the error.

This appendix contains the majority of error messages you are likely to come across, but is not exhaustive. However, if you encounter an error message not listed here, it may well have been generated by one of your applications; consult the documentation for the relevant application.

Remember that the great majority of operating system external commands have a /H(elp) option which displays a screen of help text on the command.

Reformatting

In some cases this appendix suggests reformatting and/or repartitioning as a remedy for some hard disk error conditions. Bear in mind that such action is an *extreme corrective measure*, because it destroys ALL the files within the partition. If possible, back up all the files within the partition first.

Performing a system reset

Error conditions that require you to perform a system reset, by pressing **Ctrl+Alt+Del** if global reset is enabled, or invoking the REBOOT command, are rare. If such a condition occurs make every effort to leave your program normally before resetting your system. If you have no alternative but to reset your system, try to close any open files before doing so.

Abort, Retry, Ignore, Fail ?

Origin	Several commands
Explanation	Typically this message follows an error in a read or write operation. You can respond by typing A, R, I or F. If the reason for the error is obvious (such as no diskette in the target drive) then correct the error and type R to try again. A (for abort) stops the current operation. I results in the next stage of the current operation continuing, as if the error had not been detected. F lets the current operation know that there is an error, and it may therefore continue or stop. In both the last two cases there is a risk of data being corrupted.

Access denied
Access denied: file ...
Access denied on source directory
Access to file ... denied

Origin	Several commands.
Explanation	The file is probably either a read-only file or a password-protected file. On non-secure systems read-only files can be changed to read-write via the ATTRIB command. To access a password-protected file you need to type the filename followed by a semicolon and then the password. On secure systems, your access rights to a file or directory depend on the permissions set by the owner.

Attribute A,D,H,R or S expected to follow

Origin	XDIR
Explanation	You need to state an attribute explicitly when you type the command line. Retype the command line and include a letter to name the attribute. You can find out what the letters stand for by typing the command name followed by the /H(elp) option.

Attribute A,H,R or S expected to follow

Origin	ATTRIB, XCOPY
Explanation	You need to name the attribute explicitly when you type the command line. Retype the command line and include a letter to name the attribute. You can find out what the letters stand for by typing the command name followed by the /H(elp) option.

BACKUP file sequence error

Origin	RESTORE
Explanation	You have probably inserted a diskette from the wrong set of diskettes. If you have two sets of backup diskettes you may have restored from diskette 1 in the first set and then tried diskette 2 from the second set. Try again with the correct diskette.

Bad command or parameter; press ? for help

Origin	SID
Explanation	You have typed an incorrect command line. If you enter ? you will get a list of the valid SID commands. Typing ?? gives a more detailed listing with command syntax.

Bad file name

Origin	SID
Explanation	A filename in an E, R, or W command is incorrectly specified.

Bad media type

Origin	Any application
Explanation	The diskette you have inserted into the diskette-drive is not of the expected format. Try another diskette.

BAD OPCODE interrupt from *processname* at *nnnn:nnnn*
Type any key to stop process

Origin	System error
Explanation	A program has tried to execute an invalid instruction. Type any key and then check the program for errors.

Bad or missing file

Origin	Operating system startup procedure.
Explanation	The operating system either cannot find or cannot read the cconfig.sys file on the root directory of your startup disk. Alter the cconfig.sys file and then restart the system.

Batch file ... missing retry (Y/N)?

Origin	Batch files
Explanation	The operating system was unable to find the batch file that it was in the midst of processing. The batch file may have been deleted by one of the commands it contained. Check your backup copy and correct the command that caused the problem.

Batch files nested too deep

Origin	Batch files
Explanation	One batch file can be called from within another batch file using the CALL command. The first batch file is then said to be <i>nested</i> within the second. This error message tells you that too many CALL statements have been made within the file. Check the batch files and correct the command causing the error.

Binary reads from a device are not allowed

Origin	COPY
Explanation	You have attempted to copy from a device using the binary switch /B. Re-enter the COPY command without the /B switch.

Board faulty

Origin	System error
Explanation	A fault has been detected after initialization of an ARNET multiport board. Contact your hardware supplier.

Board not found

Origin	System error
Explanation	In adding an ARNET multiport board you have not used any of the valid port addresses. Check in Appendix C for the correct address to use.

Cannot access drive

Origin	BACKUP
Explanation	The target disk may not be inserted correctly, or it may be corrupt.

Cannot close

Origin	SID
Explanation	The disk file written by a W command cannot be closed.

Cannot copy to/from a reserved device

Origin	REPLACE
Explanation	REPLACE cannot be used to copy files to devices such as modems or printers. Use the COPY command or the PRINT command if either of these are appropriate.

Cannot open ... (device)

Origin	FIND
Explanation	The FIND command cannot be used with devices. It is only used with files kept on disk. Use the COPY command to copy the relevant files to disk first, and then perform the FIND.

Cannot XCOPY to/from a reserved device

Origin	XCOPY
Explanation	You have attempted to XCOPY to a device such as a modem or printer. XCOPY should only be used for copying to disks. Use the COPY command instead.

Can't load DOS file

Origin	Operating system startup procedure
Explanation	The file may be missing or corrupt. Start the operating system from your startup diskette. You will then need to copy the system files from that startup diskette back to the hard disk or diskette that failed, using SYS.

Command or filename not recognized

Origin	Any command
Explanation	You have typed an invalid command. Check the spelling and re-enter the command, or read Chapter 11, "Commands," for a full list of valid commands. Also check the path to ensure that the required program, if correctly spelt, should be found.

Could not create new volume label on destination disk

Origin	BACKUP
Explanation	The disk is probably physically write-protected. Remove the write-protection.

Could not delete destination disk volume label

Origin	BACKUP
Explanation	The disk is probably physically write-protected. Remove the write-protection.

Could not load FORMAT

Origin	BACKUP
Explanation	BACKUP tried to run FORMAT because it detected that the target diskette was unformatted. Ensure that the two files BACKUP.COM and FORMAT.COM can both be accessed before you repeat the command. Alternatively repeat the BACKUP command with a blank formatted disk in the target drive.

Could not update destination disk backup file
Could not update destination disk control file

Origin	BACKUP
Explanation	Your disk may be corrupted.

Could not update log file

Origin	BACKUP
Explanation	The BACKUP log file (if you have chosen to create one) is kept on the root directory of the disk from which you start the operating system.

Current path invalid

Origin	FIND
Explanation	Retype the command with the correct path.

Cyclic copy not allowed

Origin	XCOPY
Explanation	You have attempted to copy subdirectories using the /S switch. However, some of the subdirectories that would be copied are named as the target directories to receive the copy. Copy to different directories, or to a different drive.

Data error

Origin	Any application
Explanation	There is an error accessing the device specified. Check the device (for example make sure it is connected properly). If the device being accessed is a disk drive (hard or diskette), this error implies that data on the disk is corrupt. Make backups of all non-corrupt files on the disk, and then either reformat the disk, or use a different disk.

Date is not valid

Origin	BACKUP, RESTORE
Explanation	The command cannot be executed with the date you have specified. The date may be typed wrongly, or it may be in the wrong format. Retype the command with a valid date. You may find the DATE command helpful. Using it you can display the current date in the preferred format.

Date must also be specified

Origin	BACKUP
Explanation	You need to include the relevant date when you specify a time in BACKUP.

Delete failed

Origin	XDEL
Explanation	Some or all of the files may be password-protected or read-only. See Chapter 8, "Protecting your information," for details of handling protected files.

Destination disk is full

Origin	RESTORE
Explanation	There is no further room for files on this disk. Restore the files to a different disk.

Destination file content lost before copy

Origin	COPY
Explanation	You have used the + feature within COPY to try to concatenate a file onto itself. Copy the original file to a different filename. Then use the + feature to create a new file made from the original file concatenated with the renamed copy.

Destination must be a valid disk drive

Origin	BACKUP
Explanation	Valid drives are the physical diskette and hard disk drives. The destination drive must be different from the source drive. Retype the command with the correct drives.

Destination path not found

Origin	BACKUP, RESTORE
Explanation	You have probably mistyped the path. Try typing the command path again.

Device does not exist

Origin:	PRINTMAP
Explanation	The physical device selected does not exist. Use PRINTMAP /H for information on how to display valid devices.

Disk drives or diskette formats are incompatible

Origin	DISKCOMP
Explanation	The two types of diskette to be compared are not the same. For example, you may be trying to compare a 1.2Mbyte diskette with a 360Kbyte diskette.

Disk full

Origin	Any command
Explanation	Erase some unwanted files from your disk and try again.

Disk read error

Origin	SID
Explanation	The disk file specified in an R command could not be read properly.

Disk write error

Origin	SID
Explanation	A disk write operation could not be successfully performed during a W command, probably due to a full disk.

Divide interrupt from *processname* at *nnnn:nnnn*
Type any key to stop process

Origin	System error
Explanation	A program has tried to execute an illegal divide operation, such as a divide-by-zero. Type any key and then check the program for errors.

DMA transfer error: Use ALIGN option in CCONFIG.SYS

Origin	System error
Explanation	An application that uses a DMA controller directly (such as copy-protection or tape-streamer software) has attempted to transfer data to an area of memory it does not own. Set the cconfig.sys setting for ALIGN to ON, and reboot. Alternatively, try using the ALIGN setting in PIFED.

Drive not ready

Origin	Any command
Explanation	The diskette is not correctly in the drive, or the drive door is still open.

Environment error

Origin	SET
Explanation	The operating system cannot find the current environment and so cannot change it.

Environment full**Origin** SET**Explanation** You have probably tried to add one too many variables to the environment.

Display the current variables by typing the SET command without any options, and then delete any environment variables that you do not need. Alternatively, review your ENVSIZE settings in **cconfig.sys**, or use the /E switch in COMMAND to increase the size of your environment.

Error on file ...**Origin** PRINT**Explanation** PRINT has discovered an error in a file that it was trying to print.

Attempt to recover the file to a different area of the disk.

Error reading configuration file**Origin** FILELINK**Explanation** Your configuration file cannot be processed. Check that the format of the file is correct.**Error reading filelist . . .****Origin** Several commands**Explanation** The "filelist" file you specified cannot be processed.

Check the format of your filelist. It must be in ASCII text format, use EDITOR to create it, not a word processor. Each filename in the list must begin on a new line. If you still have problems check the integrity of your disk using CHKDSK.

Error reading from standard input

Origin	SORT
Explanation	Standard input could not be read. Check your source file or device for errors.

Error reading system area

Origin	FDISK
Explanation	This message implies a hardware failure. Contact your dealer.

Error reading system files

Error writing system files

Origin	FORMAT, SYS
Explanation	Format has been unable to copy system files to a newly-formatted disk. Check that the drive door is closed. If it is you may have a defective diskette.

Failed to change mode of printer . . .

Origin	MODE
Explanation	A MODE change could not be carried out on the printer. Check that the printer is connected properly, and is online.

Failed to get current path on drive...(OS error code ...)

Origin	Several commands
Explanation	This is probably an error on the disk in the drive mentioned or a problem with the disk drive itself.

File already open

Origin	Several commands
Explanation	The file cannot be accessed by the command you specified because the file is already open. This may mean it is in use by another program already in memory. Close the file and try again.

Filename too long

Origin	Several commands
Explanation	The operating system supports filenames up to eight characters with an optional filename extension up to three characters. This extension is separated from the filename by a period (.). Retype the command with a shorter filename.

File not erased

Origin	ERASE, ERA
Explanation	The file may be a password-protected file or a read-only file. Set the appropriate access permissions with the XATTRIB command. Or retype the command but add the correct password after the file specification, or use the ATTRIB command to change a read-only file into a read-write file.

File not found

Origin	Several commands
Explanation	The command cannot find the file specified in your command line. Check the spelling of the filespec. If the filespec was spelt correctly, use DIR or XDIR to check the contents of other directories for the location of the specified file.

Files can only be given to an individual (... is a group)

Origin	XATTRIB
Explanation	You must specify a valid user name when changing the ownership.

File Sharing Conflict

Origin	Any application
Explanation	The action you have requested conflicts with the actions of another user. (See Chapter 15, "Running DOS applications," for information on running multiple copies of the same application.)

FORMAT not allowed on fixed media

Origin	FORMAT, BACKUP
Explanation	The command you have issued only works with diskettes. Use FDISK to format hard disks.

Format unsuccessful

Origin	FORMAT
Explanation	The diskette may be corrupted. Try again. If the message recurs, your diskette is probably unusable. Consult your dealer.

I/O error . . .

Origin	Any application
Explanation	An Input/Output error occurred while accessing the requested device. In the case of a diskette-drive, this may mean a corrupt diskette.

IBMBIO.COM read error

Origin	Diskette system loader
Explanation	The programs used to start up the operating system from diskette are corrupt or in the wrong order. Start the system from a different startup diskette. If you want to use the failed startup diskette again you will need to reformat it using the /S (copy system files) switch within FORMAT.

WARNING	Reformatting a diskette destroys all the files originally stored on it.
----------------	---

Illegal drive in Path Specification

Origin	Several commands
Explanation	You specified a drive that is not assigned in your system, or you mistyped the drive specification. Retype the command with a valid drive.

Incompatible format for drive

Origin	DISKCOPY
Explanation	The diskette in the drive is not in the expected format for the drive.

Incorrect version of operating system

Origin	Several commands
Explanation	The version numbers are different from the operating system running on your computer and the file containing the utility that you have just tried to load. The best solution is not to keep any old version of the operating system on your disks.

Input redirection already active

Origin	Command line instructions
Explanation	You have tried to start input/output redirection when it is already happening.

Insert SYSTEM diskette in A: then PRESS Ctrl/Alt/Del...

Origin	Operating system hard disk system loader
Explanation	This message follows any error message that arises from the hard disk system loader. The effect is to get you to restart the operating system from your startup diskettes.

Insufficient memory

Origin	SID
Explanation	There is not enough memory to load the file specified in an R or E command.

Internal error code ...

Origin	Various commands
Explanation	The system may have run out of internal resource space. Consult your system administrator with a view to having the system memory pool or queue buffer space and flags increased using SETUP.

Invalid attempt to both set ... and remove

Origin	PASSWORD
Explanation	You have attempted to set and reset an option at the same time. Refer to the PASSWORD command in Chapter 11, "Commands," for the permitted options and then retype the command.

Invalid baud rate

Origin	MODE
Explanation	The baud rate specified is not one of the standard baud rates. See the description of MODE in Chapter 11, "Commands." Retype the command with a standard baud rate.

Invalid break switch (use ON or OFF)

Origin	The cconfig.sys file in the root directory of your startup disk.
Explanation	Use the SETUP program to set BREAK to ON or OFF , or directly modify the cconfig.sys file, then restart the computer. Refer to the description of cconfig.sys in Chapter 16, "Customizing your system," for details of the BREAK command.

Invalid characters in label

Origin	LABEL
Explanation:	Create a different label, using only letters and numbers.

Invalid character in password

Origin	PASSWORD
Explanation:	Create a different password, using only letters and numbers.

Invalid combination of options ...

Origin	REPLACE, XCOPY
Explanation	Certain options conflict if they are selected together. Refer to the command explanations in Chapter 11, "Commands," for the available options. Type a new command that avoids the conflict.

Invalid command in CCONFIG.SYS file

Origin	Startup procedure
Explanation	The operating system does not recognize one of the commands in the cconfig.sys file on the root directory of your startup disk. Remove the invalid command from cconfig.sys ; the valid commands are described in Chapter 16, "Customizing your system." There is no need to restart the computer.

Invalid country code

Origin	The cconfig.sys file in the root directory of your startup disk.
Explanation	Alter the original cconfig.sys file and then restart the operating system. See the description of cconfig.sys in Chapter 16, "Customizing your system," for a list of the supported country codes.

Invalid date specified

Enter date: _____

Origin	DATE, TOUCH
Explanation	Enter the date in the correct format for your country.

Invalid destination pathname

Origin	REPLACE
Explanation	Retype the command with the correct path for the destination of the copied files.

Invalid drive specified

Origin	Several commands
Explanation	You specified a drive that is not assigned in your system, or you mistyped the drive specification. Retype the command with the correct drive specification.

... invalid filename

Origin	Several commands
Explanation	Retype the command with the correct filename.

Invalid file specification

Origin	PRINT
Explanation	Retype the command with the correct path and filename.

Invalid hard disk system partition

Origin	Hard disk system loader
Explanation	The system loader is unable to read information in the Partition Table. Restart the operating system from diskette. Use the FDISK utility to recreate the partition on the hard disk.

WARNING	Using FDISK will destroy all the data on that partition of your hard disk.
----------------	--

Invalid last drive character (use A...Z)

Origin	The cconfig.sys file in the root directory of your startup disk.
Explanation	Use the SETUP program to amend the LASTDRIVE entry in cconfig.sys or modify cconfig.sys directly and restart the operating system.

Invalid lines per inch

Origin	MODE
Explanation	You must specify either 6 or 8 lines per inch for your printer in the MODE command.

Invalid line width

Origin	MODE
Explanation	You must specify 80 or 132 characters per line for your printer in the MODE command.

Invalid number of buffers

Origin	The cconfig.sys file in the root directory of your startup disk.
Explanation	Amend the BUFFERS entry in cconfig.sys using the SETUP program or modify cconfig.sys directly, and then restart the operating system.

Invalid number of data bits

Origin	MODE
Explanation	The databits setting specified is not one of the standard settings.

Invalid number of parameters

Origin	Several commands
Explanation	You have typed too many, or conflicting, options.

Invalid number of stop bits

Origin	MODE
Explanation	The stopbits setting specified is not one of the standard settings.

Invalid option - decimal number expected

Origin	SORT
Explanation	The /+n option of SORT stands for the on-screen column number at which you want the sort to occur. This number must be given as an ordinary decimal number and not as a hexadecimal.

Invalid option ... in parameter ...

Origin	Several commands
Explanation	You have typed an option that the operating system does not recognize. Refer to the command description in Chapter 11, "Commands," for the correct options and then retype the command.

Invalid parameter

Origin	Several commands
Explanation	You have typed an option that the operating system does not recognize. Refer to the command description in Chapter 11, "Commands," for the correct options and then retype the command.

Invalid parameter - missing option

Origin	FIND
Explanation	You have typed an option that the operating system does not recognize.

Invalid parity setting

Origin	MODE
Explanation	The parity setting specified is not one of the standard parities.

Invalid partition table

Origin	Hard disk system loader
Explanation	More than one partition is marked as bootable (that is able to start the operating system), a non-existent partition is marked as bootable, or there is a hard disk failure. Restart the operating system from diskette. Use the FDISK utility to set the bootable partition.,

Invalid password

Origin	Several commands
Explanation	The password you have typed is not correct.

... invalid path

Invalid path or path not found

Invalid path specified: check path

Origin	Several commands
Explanation	Retype the command with the correct path.

Invalid switch .. (PRINT already installed)

Origin	PRINT
Explanation	The option you chose cannot be set because PRINT is already installed in memory. Carry on without the option, or restart the operating system and load PRINT again.

Invalid time specified

Enter time: ____

Origin	TIME, TOUCH
Explanation	You entered the time in an incorrect format. Enter the time in the format as shown in the screen prompt.

... is not a valid user login name or group name

Origin	Various commands with security enabled
Explanation	You have typed a user or group name not recognized by the operating system. Enter a valid name.

Label ... not found

Origin	Batch file and cconfig.sys processing
Explanation	A GOTO statement in the batch or cconfig.sys file refers to a label that does not exist. Correct the relevant file.

Line too long in file

Origin	FIND
Explanation	You have tried to use FIND on a non-text file. FIND is only designed for text files.

Loading conflict

Origin	Several commands
Explanation	There is probably not enough memory for the task you have specified. BANK may have been set to OFF. Free some memory by removing memory-resident programs that you do not need, or by reducing the amount of memory occupied by memory disks. Alternatively, make sure BANK is set to ON.

Log file path not found

Origin	BACKUP
Explanation	You have probably mistyped the path for the log file. Try again.

Mono display adapter not found

Origin **MODE**

Explanation The MONO display mode that you have specified is not supported by your display adapter.

New login password could NOT be assigned

Origin **PASSWORD** (secure systems only)

Explanation The original password has been mistyped, so you cannot change it.

New passwords were not identical

Origin **PASSWORD** (secure systems only)

Explanation You have not typed the identical password twice over. Retype the password from the beginning.

No backup files present on source disk

Origin **RESTORE**

Explanation You inserted a disk that contained no backup files, or you inserted a backup diskette out of sequence.

No file

Origin **SID**

Explanation The file specified in an R or E command could not be found on the disk.

No files to backup

Origin **BACKUP**

Explanation No files matching the conditions you specified were found. You may have mistyped the file specification.

No operating system

Origin	Hard disk system loader
Explanation	This message originates in software that your supplier provides with the hard disk. Start the operating system from the diskette. Use SYS to copy the system onto the hard disk and then use the FDISK utility to make the partition "bootable" (that is, capable of loading the operating system when the computer is started).

No space

Origin	SID
Explanation	There is no space in the directory for the file being written by a W command.

Non standard sector length - cannot write new boot sector

Origin	SYS
Explanation	You have typed Y when asked whether a new boot sector is to be written. However SYS cannot recognize the disk format, since the sector size is not 512 (or a multiple of 512 for a hard disk). Backup any data on the disk and reformat it.

Not enough disk space

Origin	FILELINK, REPLACE, SORT, XCOPY
Explanation	The disk has too much stored on it. Erase some unwanted files from the disk and try again.

Not enough memory
Not enough memory for ...
Not enough memory to ...

Origin	Several commands
Explanation	There is insufficient free memory for the command to work. Free some memory by creating a smaller memory disk, or by removing memory-resident programs or device drivers that you do not need. Then try the command again.

Not ready error

Origin	Any application
Explanation	The device you are trying to access is not responding. In the case of a diskette-drive, this usually means there is no diskette in the drive, or the drive door is not shut. With hard disks, this error may mean a hardware failure. Check your diskette is inserted correctly into the diskette-drive, and the door is shut. For hard disks, contact your dealer.

Operand

Origin	SID
Explanation	An attempt was made to assemble a command with an ambiguous operand. Precede the operand with the prefix "BYTE" or "WORD."

Operating system error code
Operating system error number

Origin	Several commands
Explanation	If you get this error message, make a note of the situation in which it occurred, and contact your dealer.

Operating system error: This program can not run on a non-graphics console

Origin	System error
Explanation	You have tried to run a program that has been written specifically for graphics terminals and so cannot run on the terminal(s) in your system.

Operating system error: This program can not run on a serial console

Origin	System error
Explanation	You have tried to run a program that has been written specifically for MS-DOS and cannot run on dumb (non-PC-emulation) terminals.

Operating system load error

Origin	Hard disk system loader
Explanation	The computer cannot read the operating system's loader file. Restart the operating system from diskette. Try to restore the system using SYS. If this does not work, use BACKUP to save the files on the partition onto diskette. Use the FDISK utility to delete and then recreate the partition. Then use RESTORE to recover your files.

Out of GENSYS flags

Origin	System error
Explanation	The operating system has run out of system flags. Reconfigure the system with more flags, using the FLAGS command in cconfig.sys.

Output redirection already active

Origin	Command line instructions
Explanation	You have tried to start input/output redirection with it already happening.

Overflow interrupt from *processname* at *nnnn:nnnn*

Type any key to stop process

Origin	System error
Explanation	A multiply or divide operation in a program has generated an overflow. Press any key to abort the process and continue using the system.

Page fault interrupt from *processname* at *nnnn:nnnn*

Type any key to stop process

Origin	System error
Explanation	A program has tried to access memory that does not exist or is write-protected. Press any key to abort the process and continue using the system.

Parameter not recognized

Origin	BACKUP, MODE, RESTORE
Explanation	You have typed an invalid option. Refer to the relevant command description in Chapter 11, "Commands," for the permitted options.

Parameters not compatible

Origin	FORMAT
Explanation	The options that you have typed contradict each other. Refer to the entry for the FORMAT command in Chapter 11, "Commands," and then retype the command.

Parameters not supported for drive

Origin	FORMAT
Explanation	The options you have typed are not supported on the diskette drive you specified.

Parity error from *processname* at *nnnn:nnnn*

**** MACHINE IS HALTED ****

Origin	System error
Explanation	A memory operation has generated a parity interrupt, not necessarily in the named process at the named address. You have a hardware error. Note the circumstances of the error and contact your dealer. Switch off your machine, then switch it on again to resume work.

Path not found

Origin	Several commands
Explanation	The path you have typed is incorrect. Retype with the correct path.

Path too long

Origin	PIFED
Explanation	You have tried to specify a path name that is too long. Retype with the correct path.

Path was too long to backup - skipping over

Origin	BACKUP
Explanation	Your hierarchy of subdirectories is too complex. Simplify the arrangement of subdirectories on your disk and then try BACKUP again.

Physical Media Error

Origin	Several commands
Explanation	The operating system was unable to read from or write to a disk. The disk has probably been corrupted. Copy as many files as you can from the corrupt disk to another disk and then reformat or dispose of the first disk. This error message also appears if you have PIFEDed an application for a resource that is not available, such as a graphics screen. In this case, make sure you have the correct configuration, then rerun the application.

Printer busy

Origin	PRINT
Explanation	You have tried to access a printer that is already in use. Wait until the printer is free and try again.

Print queue full

Origin	PRINT
Explanation	If the printer is printing, try again later. Otherwise, delete a file that you do not need from the print queue and then try adding your file again.

Printer out of paper

Origin	Any application
Explanation	The printer you are attempting to access is out of paper. Refill the paper tray, and try printing again.

Program terminated normally

Origin	SID
Explanation	The program running under SID completed, or was terminated by a Ctrl+Break.

Protection interrupt from *processname* at *nnnn:nnnn*
Type any key to stop process.

Origin	System error
Explanation	A non-specific protection interrupt has been trapped at address <i>nnnn:nnnn</i> . The process may have attempted to write to read-only memory. Press any key to abort the process and continue.

Protection level not changed file has no password

Origin	PASSWORD
Explanation	This message informs you that no password is attached to the file you have named in the command line.

Read error on file

Origin	FIND
Explanation	Your disk may be corrupted. Try again with a backup copy.

Resident portion installed

Origin	MODE
Explanation	A portion of the MODE.COM file remains in memory once MODE is invoked, slightly increasing the amount of space occupied by the operating system. This portion is called the <i>resident</i> portion.

Sector not found

Origin	Any application
Explanation	The requested sector on the disk is not present, or is corrupt. With diskette drives, this usually means that the diskette in the drive has not been formatted.

Seek error

Origin	Any application
Explanation	The disk drive being accessed was physically unable to move its read/write heads to the requested location. This generally means a disk controller hardware failure. Contact your dealer.

SETUP type incorrect

Origin	SETUP
Explanation	You have installed an ARNET card device driver without configuring the system for the card. Use SETUP to identify the correct multiport card.

Source and destination cannot be the same file

Origin	COPY, REPLACE, XCOPY
Explanation	You have named the same file as both the source file and the destination file for the copying operation.

Source and destination drives are the same

Origin	BACKUP, RESTORE
Explanation	The destination drive must be different from the source drive (the drive containing the disk to be backed up or restored). Retype the command with different source and destination drives.

Source drive must be specified

Origin	BACKUP, RESTORE
Explanation	BACKUP and RESTORE need both the source drive (the drive containing the disk to be backed up or restored), and the destination drive to be included in the command you type. These drives must be different drives. Retype the command with the correct drives specified.

Source file . . . is not a valid source EXE file

Origin	EXE2BIN
Explanation	The file you tried to convert is not a valid executable type file.

Source path not found

Origin	BACKUP, RESTORE
Explanation	You have probably mistyped the path. Try again.

Specified path is too long

Origin	Several commands
Explanation	You have tried to specify a path name that is too long. Retype with the correct path.

Specify source AND destination drives

Origin	BACKUP, RESTORE
Explanation	BACKUP and RESTORE need both the source drive (the drive containing the disk to be backed up or restored), and the destination drive to be included in the command you type. These drives must be different drives. Retype the command with the correct drives specified.

Switch value out of range

Origin	PRINT
Explanation	The value chosen for the option is out of the allowed range. Refer to the description of PRINT in Chapter 11, "Commands," for the correct ranges for each option.

System files not found

Origin	FORMAT
Explanation	You specified the /S switch within FORMAT but the operating system cannot find the system files to copy onto the newly formatted disk. Retry the command with a copy of FORMAT that is on the disk from which you start the operating system.

This program suspended while other program(s) in graphics:
No memory or disk buffer available to save graphics screen

Origin	System error
Explanation	Not enough graphics buffers are available to run the number of graphics-based applications required. Use SETUP to increase the number of graphics save buffers.

Time is not valid

Origin	BACKUP, RESTORE
Explanation	You have probably mistyped the time. Retype the command with a valid time. You may find the TIME command helpful. Using it you can display, and if necessary, change the current time.

Too few parameters

Origin	MODE
Explanation	You have not typed in enough information. Refer to Chapter 11, "Commands," for a description of the MODE command and its various formats and parameters.

Too many files open

Too many open files

Origin	Several commands
Explanation	You need to increase the number of files that can be open at one time, using SETUP.

Too many files to sort

Origin	XDIR
Explanation	There is not enough room for XDIR to sort and then display all the files you have specified. Sort files from a subdirectory so that XDIR is sorting fewer files.

Too many parameters

Origin	Several commands
Explanation	You have specified too many, or conflicting, options. Refer to the description of the relevant command in Chapter 11, "Commands."

Unable to backup due to sharing conflict

Origin	BACKUP
Explanation	The file you specified is currently being used by another process. Retry the command when the file is not being used.

Unable to convert. Initial IP of . . . must be 0 or 100h

Origin	EXE2BIN
Explanation	For an .exe file to be converted to a .com file, its entry point must be at offset 100h within its segment. For an .exe file to be converted to a .bin file, its entry point must be at absolute offset 0 in its memory usage. If either of these cases is untrue, the conversion cannot be carried out.

Unable to convert. Source file . . . has a stack declared

Origin	EXE2BIN
Explanation	You cannot convert a .exe file to either a .com or .bin file if the program has a stack statement declared.

Unable to convert. Source file ... requires relocation fixups

Origin	EXE2BIN
Explanation	The destination file is to be a .com type, but its header contains relocation items which must be performed.

Unable to convert. Source file . . . requires too much memory

Origin	EXE2BIN
Explanation	The destination file is to be a .com type but it requires more than 64 Kbytes to run.

Unable to create directory

Origin	FILELINK, MKDIR, XCOPY
Explanation	Your disk is probably full, or it is physically write-protected, or corrupt. You can also get this message if you try to create a directory with the same name as an existing directory. Use DIR to find out the amount of free space on the disk, and then delete unwanted files. Check the disk for write-protection. Try creating a different directory name, or create the directory on a different drive.

Unable to delete existing volume label

Origin	LABEL
Explanation	The diskette is probably physically write-protected.

Unable to find start of program code

Origin	EXE2BIN
Explanation	The source file contained a valid .exe header but no valid program code.

Unable to open or create destination file

Origin	EXE2BIN
Explanation	The program was unable to access the specified destination file. This is probably because there is already a read-only file of the same name.

Unable to open source file

Origin	EXE2BIN
Explanation	The specified source file does not exist. Check the name of your source file.

Unable to read BACKUP file

Origin	RESTORE
Explanation	There is a problem with the diskette you have inserted. It may be the wrong diskette, or it may have been inserted out of sequence, or corrupted. Try again with the correct diskette.

Unable to read relocation information

Origin	EXE2BIN
Explanation	The segment fixup information in the .exe header of the source file is not valid.

Unable to restore due to sharing conflict

Origin	RESTORE
Explanation	The file you specified is currently being used by another process. Retry the command when the file is not being used.

Unable to send mail to that user at this time

Origin	MAIL
Explanation	You have elected to send mail to a user whose in-tray is currently unavailable. Try again in a few minutes.

Unable to set volume label

Origin	LABEL
Explanation	The diskette is probably physically write-protected. Remove the write-protection and try again.

Unable to write to destination

Origin	RESTORE
Explanation	The destination disk is probably physically write-protected. Remove the write-protection and try again.

Unable to write to destination file

Origin	EXE2BIN
Explanation	The program was able to open, but could not write to, the destination file. This is probably due to a hardware malfunction.

Unexpected interrupt

Origin	System error
Explanation	The system has trapped an interrupt it cannot deal with.

Unknown error code

Origin	Several commands
Explanation	If you get this error message, make a note of the situation in which it occurred, and contact your dealer.

Unknown switch

Origin	Several commands
Explanation	You have specified an option that does not exist. Refer to the description of the command in Chapter 11, "Commands," for a list of the valid options. Then retype the command.

Unrecognized display mode

Origin	MODE
Explanation	You have probably mistyped the display mode. The MODE command description in Chapter 11, "Commands," contains a listing of the display options.

Unrecoverable read error on SOURCE diskette
Unrecoverable read error on DESTINATION diskette
Unrecoverable format error on DESTINATION diskette
Unrecoverable verify error on DESTINATION diskette

Origin	DISKCOMP, DISKCOPY
Explanation	The diskette is probably corrupt. Use the backup copies of your files.

Use another boot disk

Origin	Diskette system loader
Explanation	The programs used to start up the operating system from diskette are corrupt, or in the wrong order. Start up the operating system from a different startup diskette. If you want to use the failed startup diskette again you will need to reformat it using the FORMAT command with the /S (copy system files) switch.

WARNING	Reformatting a diskette destroys all the files originally stored on it.
----------------	---

Verify error

Origin	FORMAT
Explanation	You probably have a corrupt disk. Consult your dealer.

Verify error at s:o

Origin	SID
Explanation	The value placed in memory by a Fill, Set, Move, or Assemble command could not be read back correctly, indicating bad RAM or attempting to write to ROM, or non-existent memory at the indicated location.

Write error

Origin	FORMAT
Explanation	You probably have a corrupt disk. Consult your dealer.

Write protect critical error

Origin	Any application
Explanation	The diskette to which you are attempting to write is physically write-protected. Remove the write protection or use a different, non-write-protected diskette.

Wrong boot sector format for SYS

Origin	SYS
Explanation	You may have typed N when asked whether a new boot sector is to be written. SYS only copies files to a disk with an operating system boot sector.

Wrong OS version**Wrong version of operating system**

Origin	Various commands
Explanation	The version numbers are different between the operating system running on your computer and the file containing the utility or driver that you have just tried to load. The best solution is not to keep any previous version of the operating system on your disks.

A.2 Troubleshooting

This section discusses several classes of errors, their causes, and some recovery procedures under the following headings:

- ☐ Troubleshooting checklist
- ☐ Diskette handling
- ☐ Corrupted files
- ☐ When absolutely nothing happens
- ☐ Unresolvable errors

Troubleshooting checklist

This checklist covers some of the more common error conditions.

- ☐ Have you prepared a partition on your hard disk? You cannot access your hard disk until this is done. See Chapter 11, "Commands," for a description of the disk partitioning utility, FDISK.
 - ☐ Is your hard disk unit turned on? An external hard disk unit may have its own electrical connection.
 - ☐ Are the peripheral devices connected properly to your computer and to electrical outlets?
 - ☐ If a file is set to read-only, you cannot edit it. Use ATTRIB or XATTRIB to set the file to read-write.
 - ☐ If a file is passworded, you cannot edit it, unless you specify the password on the command line.
 - ☐ Did you specify a directory when typing the command line? By default the operating system looks for the file only in the current directory of the current drive.
 - ☐ Is the file password-protected? Refer to Chapter 8, "Protecting your information," and Chapter 11, "Commands," for information on password protection.
 - ☐ Are you running a program under the correct operating system version? Many commands run only under the operating system with which they are shipped.
-

Diskette handling

Diskette handling is the source of many errors. Here is a summary of points to check through (see also Chapter 5, section 5.2, which describes how to handle diskettes).

- ☐ Is the disk correctly inserted and the drive door closed?
- ☐ Is the disk the correct type for the drive?

Double-sided drives can read single-and double sided diskettes. Single-sided drives can access only single-sided diskettes.

Some drives are designed to access both 1.2 Mbyte diskettes and 360 Kbyte diskettes. Other drives will only access one of these types. Still others will only access 160 Kbyte and 180 Kbytes diskettes, or 3½" diskettes. The `FORMAT` command description in Chapter 11, "Commands," contains full details of diskette types.

- ☐ Is the disk physically write-protected? See Chapter 5, section 5.2, if you are unsure about write-protection.
- ☐ If the error is not caused by one of the above problems, the disk might be damaged. Use the `COPY` command (see Chapter 11, "Commands") to copy information to a new disk. Replace any files you cannot copy with the files from your backup disk. If you have not maintained a backup disk, you must recreate the missing files.

You can use the `CHKDSK` command to check the condition of your disks. The verification process is non-destructive. See Chapter 11, "Commands," for instructions on how to use `CHKDSK`.

If these directions fail to correct an error condition, you may be faced with a hardware error: contact your dealer.

Corrupted files

Files can be corrupted by all kinds of unforeseen events. Power failures are probably the most common cause.

Replace the damaged file from your backup copy. If you do not have a backup copy, you will have to recreate the file.

If files are corrupted after you run a program that you yourself have written, debug the program carefully. Your program might be overwriting other files on the disk, or corrupting the directory.

When absolutely nothing happens

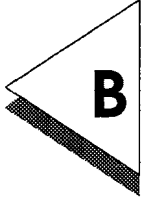
First, wait. Some programs take a long time to finish. Some errors, such as disk Input/Output errors, cause the operating system to retry the operation many times. With experience, you will know when a command is taking too long.

Unresolvable errors

If you encounter an error condition you cannot correct, contact your dealer. Be prepared to provide the following information:

- ☐ The text of any error messages you received.
 - ☐ The serial number of your operating system. (This number appears on your distribution diskette label.)
 - ☐ What your computer configuration is, that is the number of drives, quantity of memory, and any additional equipment such as printers or modems.
 - ☐ Enough information to reproduce the error. Try to recall which programs were running when the error happened. If possible, provide a diskette with a copy of the program.
-
-

Connecting a serial terminal



TO connect a serial terminal to your main console you must first make the physical connection, and then run the **SETUP** program, as described in Chapter 16, “Customizing your system,” to specify the configuration details for your terminal. This appendix describes how to make the physical connection between the terminal and your PC.

B.1 Null modem connections

To connect a serial terminal to your PC you require a special type of cable known as a “null modem” cable. This cable performs the signal translation that ensures that data is transferred correctly between the PC and the serial terminal.

A null modem cable uses a standard 25-pin connector but many PCs have a 9-pin serial port connection. If your PC has a 9-pin serial port, you require an additional conversion to perform signal swapping correctly, as pin settings for signal lines are different for 9-pin (DB9) and 25-pin (DB25) connections.

Many electronics stores supply DB9 to DB25 converters, which you can use to connect a null modem cable to the serial port on your PC. Alternatively, you can make a suitable connector by configuring a short cable with a DB9 connector at one end and a DB25 at the other. The following table shows how the pins are used for signal lines on both DB9 and DB25 connectors.

Signal	25-pin signal line setting (9 pin setting in brackets)
CD — Carrier request	8 (1)
RX — Receive Data	3 (2)
TX — Transmit Data	2 (3)
DTR — Data Terminal Ready	20 (4)
GND — Signal Ground	7 (5)
DSR — Data Set Ready	6 (6)
RTS — Ready to Send	4 (7)
CTS — Clear to Send	5 (8)
RI — Ring Indicator	22 (9)

B.2 Possible configurations

The pin connections that are required for a null modem connection between the PC and the serial terminal are indicated in the following charts. The pin connections for a 9-pin port are given in parenthesis, in case you prefer to make the connection directly from a 9-pin serial port on the PC to the serial terminal.

Minimum configuration

This configuration uses a protocol with XON/XOFF handshaking or no handshaking. The advantage of this configuration is that you only require a three-wire cable, however it is not recommended for terminals that run above 9600 baud.

25-pin PC end (9-pin)	Terminal End DB25 (9 pin)
TX — Transmit 2 (3)	RX — Receive 3 (2)
RX — Receive 3 (2)	TX — Transmit 2 (3)
GND 7 (5)	GND 7 (5)

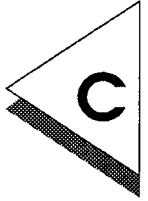
Recommended configuration

This configuration uses a protocol with DTR/DSR handshaking and requires 5 lines.

25-pin PC end (9-pin)	Terminal End DB25 (9 pin)
TX — Transmit 2 (3)	RX — Receive 3 (2)
RX — Receive 3 (2)	TX — Transmit 2 (3)
DSR 6 (6)	DTR 20 (4)
GND 7 (5)	GND 7 (5)
DTR 20 (4)	DSR 6 (6)

Multiuser DOS User Guide

Adding a multiport board

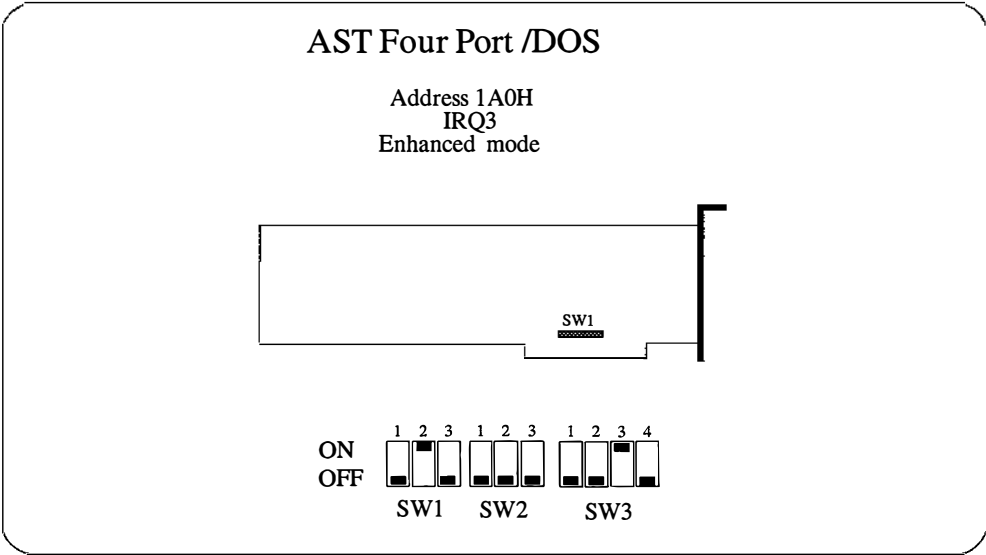
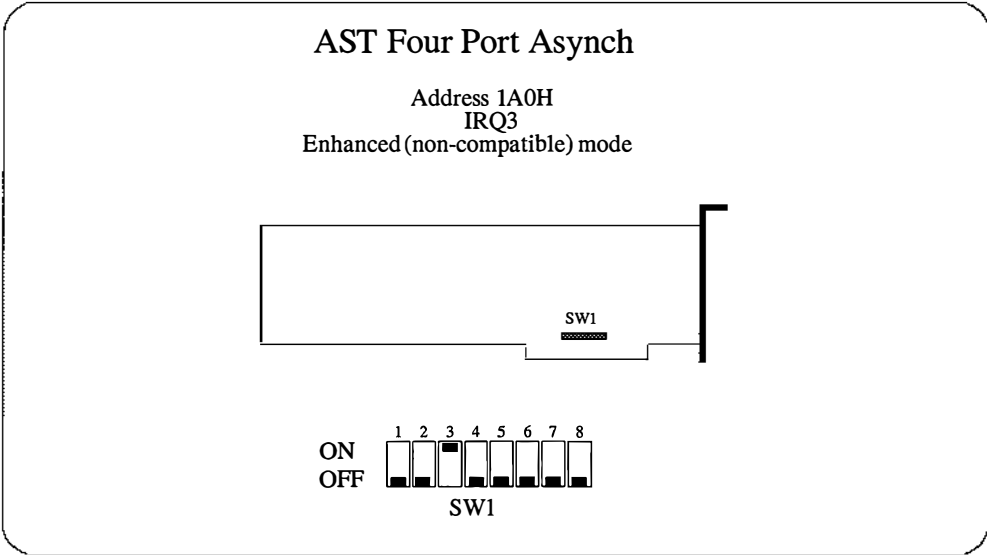


INSTEAD of using COM2 as a single serial port, you can attach a multiport board, thus expanding your range of ports. This appendix gives a number of valid switch settings for various multiport boards. It also gives advice on using the ARNET® installable device driver shipped with the operating system.

For information on all other installable device drivers, see your dealer or refer to the appropriate device driver documentation.

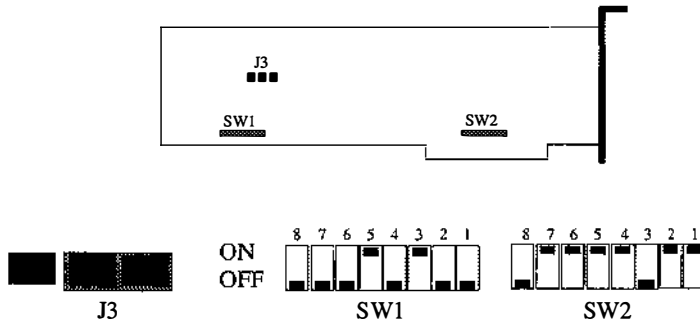
C.1 Switch settings

The following switch settings are valid:



Hostess Multiport Network Adapter - 4 Port

Address 280H
IRQ3



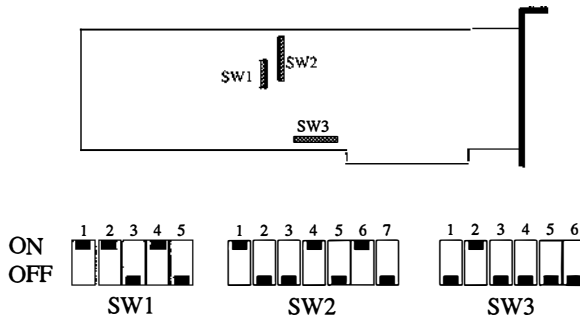
Hostess Multiport Network Adapter - 8 Port

Address 280H
IRQ3



ARNET Multiport Multi-4 and Multi-8

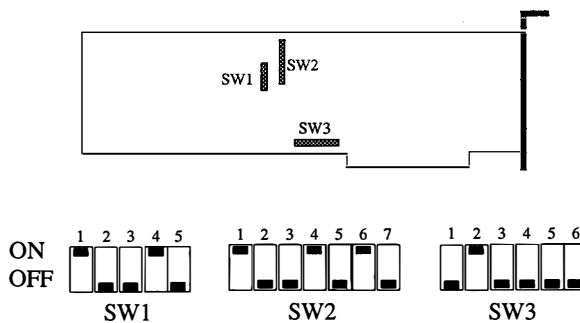
Address 280H
IRQ3
Option Address 2B0H



If you use LPT2, follow the settings shown below, because the I/O settings used by the above configuration conflict with the settings required by the parallel port LPT2.

ARNET Multiport Multi-4 and Multi-8

Address 2C0H
IRQ3
Option Address 2B0H



C.2 The ARNET device driver

Arnet.sys is an installable device driver shipped with the operating system. It enables users to use an ARNET Smartport® intelligent I/O board, and supports both the four-port and the eight-port versions of the board.

The benefit of intelligent I/O port boards is that they allow the operating system to support additional serial ports without the overheads normally associated with dumb serial port boards.

Installation

The ARNET card has to be configured for one of the following port addresses: 0280H, 02A0H, 02C0H, 02E0H or 0300H.

The driver does not require an interrupt request line (IRQ).

For the memory window, an address above the 1 Mb boundary can be used in order to leave more address space for operating system buffers above the screen RAM in the first megabyte.

To enable the driver, run SETUP and choose the generic driver option, selecting 4 or 8 ports depending on the board. All parameters are configured as for standard non-intelligent serial port boards.

Insert an entry in **cconfig.sys** with the following statement:

```
DEVICE = c:\cdos\arnet.sys
```

unless you are installing the driver in a 386 machine that uses the Intel 82385 cache controller, in which case it may be necessary to specify the /CACHE_FIX option in the **cconfig.sys** file:

```
DEVICE = c:\cdos\arnet.sys /CACHE_FIX
```

When the system is loaded, the driver signs on as follows:

```
ARNET Smartport(s) : n
```

where *n* is either 4 or 8.

Handshaking

The driver uses the CTS input line as a flow control line on output, irrespective of the protocol configured by SETUP. If CTS flow control is not desired, it has to be tied to a fixed value, such as the RTS output from the ARNET board.

When the XON-XOFF protocol is selected in SETUP, the driver uses it for auxiliary devices, printers and dumb terminals. If a port is configured as a PC terminal, however, the driver automatically assumes XPC handshake as well as RTS-CTS. If your terminal supports XPC handshaking, for example the WYSE-60, you can use a simple cable without hardware handshake lines ("Null modem cable").

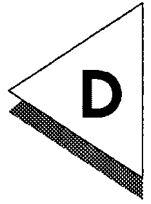
Baud rates

The ARNET hardware enforces some restrictions about the baud rates that can be configured. In each pair of ports (1st and 2nd, 3rd and 4th, and so on), if one port is programmed for 38400 baud, the other port cannot operate at 19200 baud or 150 baud; the other port must operate at 110, 300, 600, 1200, 2400, 4800, 9600 or 38400 baud. Therefore, when you configure the ports as terminals and printers, make sure you group them so that no such conflicts arise, or avoid using 38400 baud entirely. If you restrict the maximum baud rate to 19200 or less, you avoid these restrictions.

The /CACHE_FIX option

The CACHE_FIX option forces the device driver to add 2 Gigabyte to the dual port RAM address, which in turn forces the Intel cache controller to not cache data shared with the 80186 processor on the Smartport board. Use this option only if the ARNET board does not otherwise work.

Memory management



YOU can use the operating system to make efficient use of available memory on your computer. This appendix defines the different types of memory and how you can optimize your own computer's memory using various operating system commands.

D.1 What is memory?

Every computer must have memory to store programs and information, and to load and run programs. Memory used to hold programs and data while the computer is switched off is called storage memory. The other type of memory that is used to load and run programs, is called Random Access Memory (or RAM for short). It can be thought of as the computer's workspace. The more complex a task you want to do, the more RAM memory your computer needs.

Information is stored in the computer as patterns of 1s and 0s, known as bits. A group of 8 bits is called a byte. Bytes are very important units of memory; characters such as "a", "3" or "?" can be represented in one byte of memory. One thousand and twenty-four bytes is equal to one Kilobyte (or Kbyte for short). Memory is also measured in Megabytes (Mbyte). One Megabyte is equal to 1024 Kilobytes.

As modern computers contain a lot of memory, memory is usually talked about in terms of Kbytes and Mbytes rather than bytes.

D.2 Types of memory

There are several terms used to describe memory in personal computers, depending on its location (address) and usage. These different categories of memory include:

- ☐ Conventional (or base) memory
- ☐ High memory
- ☐ Upper memory
- ☐ Extended memory
- ☐ LIM (expanded) memory

Conventional memory

Conventional memory (sometimes called *base memory*) is the first 640 Kbytes of memory in the computer. It is divided into two sections:

- ❑ The operating system area in the lowest part of base memory, where your operating system and installable device drivers are commonly loaded.
- ❑ The area left over after the operating system is loaded, where your own applications and data are loaded. This is known as the Transient Program Area (TPA) or simply the *Applications Area*.

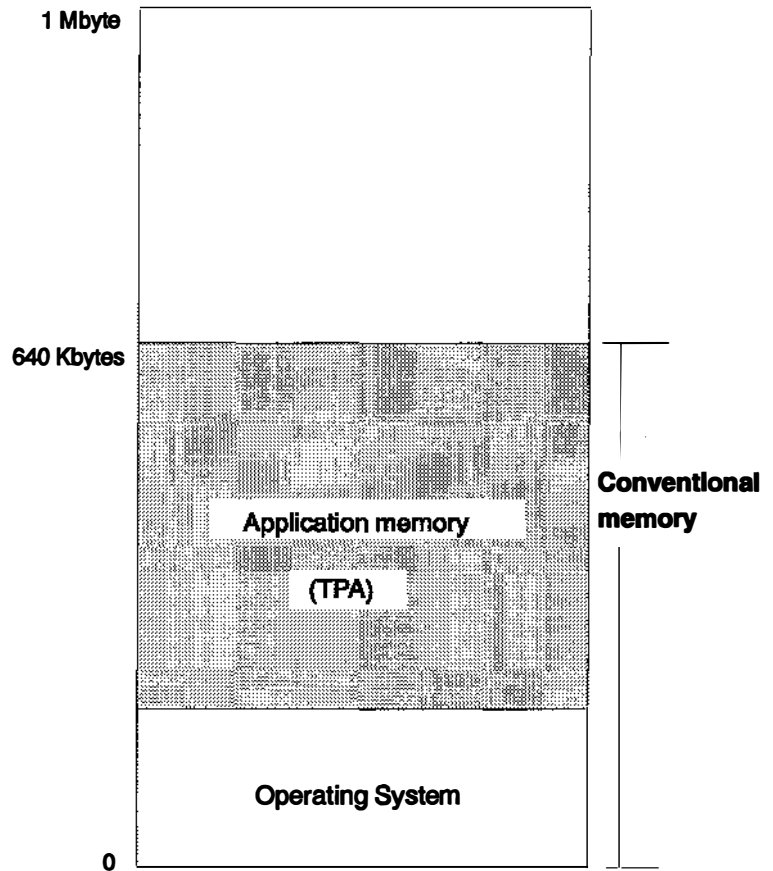


Figure D-1 Memory Map for Single User DOS

Under DOS operating systems, applications can only run in the first 640 Kbytes of memory. This limitation is sometimes called the 640 Kbytes barrier. Memory above the 640 Kbytes barrier is normally used for the video display adapter, and the ROM BIOS in particular.

The more complicated the task to be accomplished, the larger the program will be and the bigger the TPA needed. The rest of this section explains how the system can be configured to take advantage of special memory areas in your computer, to ensure that your programs get as much memory as possible.

High memory

High memory is the first 64 Kbytes of memory above the 1Mbyte barrier. It can only be used by one program, or one piece of computer code at a time and is only available on Intel 80286, i386 and i486-based computers with more than 1 Mbyte of memory installed.

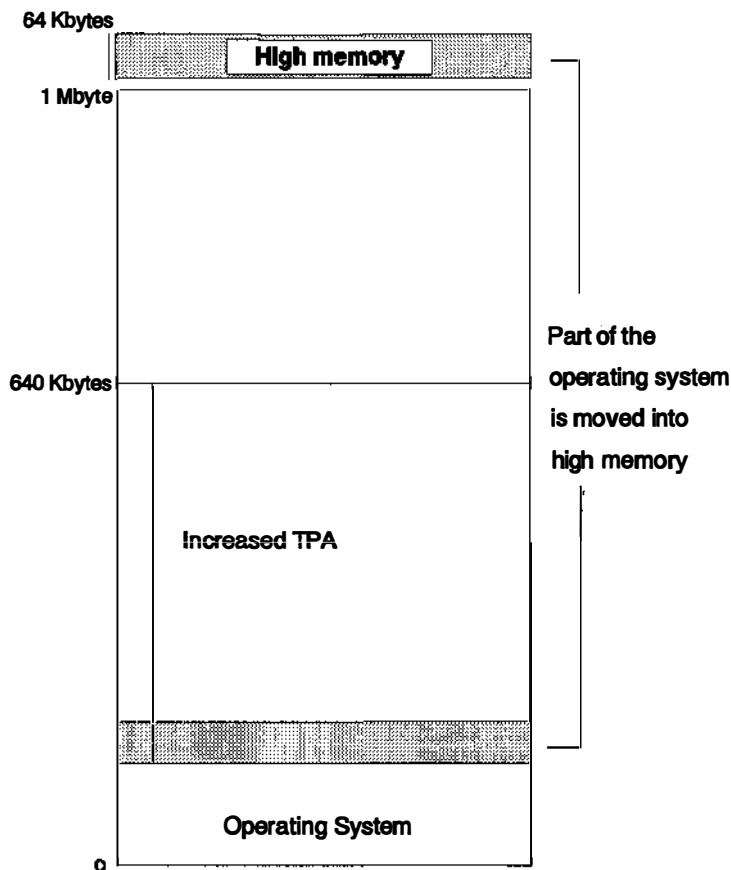


Figure D-2 Use of High Memory

The first step towards increasing the TPA taken during initialization is to move some of the operating system into the high memory area, reducing the memory address at which the TPA starts. This is done automatically.

Upper memory

Any memory addressed between 640 Kbytes and 1 Mbyte is known as *upper memory*. Upper memory is 384 Kbytes in size. Some sections of upper memory are reserved for use by various parts of your computer's hardware, like the video display adaptor, and also the ROM BIOS.

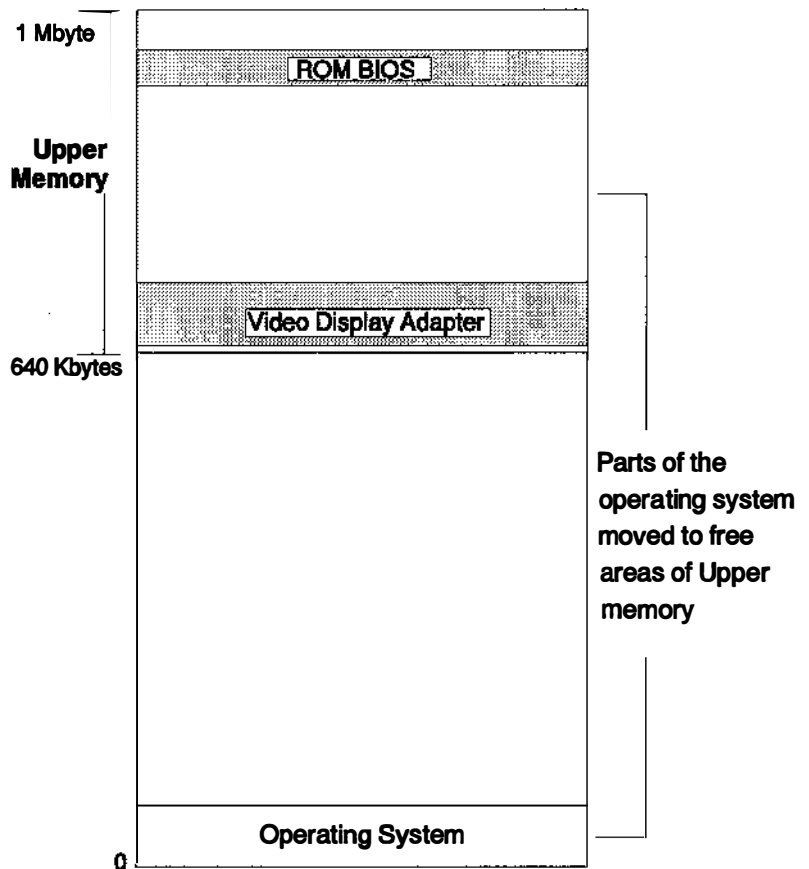


Figure D-3 Upper Memory

The operating system has an internal driver which makes use of upper memory regions. First RAM is mapped into empty regions of upper memory. Then portions of the operating system are moved from conventional memory to these areas of RAM in upper memory, increasing the amount of memory available to load applications.

The way this is accomplished is determined by the INSTALL or SETUP program option *All upper memory set to Autoscans*.

If you select *Yes* for this option, the internal driver automatically checks for free 16 Kbytes blocks of upper memory. Blocks in use are set to *Disabled*.

If you select *No*, you can then decide whether each 16 Kbyte block is going to be used, by setting its status to:

- ☐ *Disabled* - do not use this block.
- ☐ *Autoscan* - use this block the next time the operating system is started *if* it is free.
- ☐ *Enabled* - use this block even if it was not free.

The *Enabled* option allows you to override autoscanning. As well as video memory, upper memory is also used for other hardware such as network, modem and fax cards. If you use *autoscan*, the blocks of memory associated with these cards are set to *Disabled* and not used. However, you may have installed a card that is not being used and that memory would then be “wasted.” By specifically setting those blocks to be *Enabled*, you can force the operating system to use them.

WARNING Using this option can cause the system to fail when you next try to reboot. If you are unsure, use the *Autoscan* option.

Extended memory

The original IBM PC architecture allows for up to 1 Mbyte of memory in a computer. Today, many computers have more than 1 Mbyte of memory, and several operating systems use this memory as a form of memory called *extended memory*. This operating system uses extended memory for multitasking.

In the early days the only way to use extended memory was to fit a special memory expansion card inside your computer. Now,

using virtual 8086 mode on Intel i386- and i486-based computers, these special memory cards are no longer required.

Virtual 8086 mode is used to enable all the available extended memory (up to 4 Gigabytes) for use in running multiple DOS sessions.

Each DOS session is allocated a number of 16 Kbyte blocks when you start it; that is, the program you run in that DOS session is stored in extended memory. The blocks are remapped into the TPA so that the program can be run. This remapping is called *paging* and extended memory is sometimes called *paged memory* because it is divided up into these 16 Kbyte memory pages. The term *banked memory* is also used.

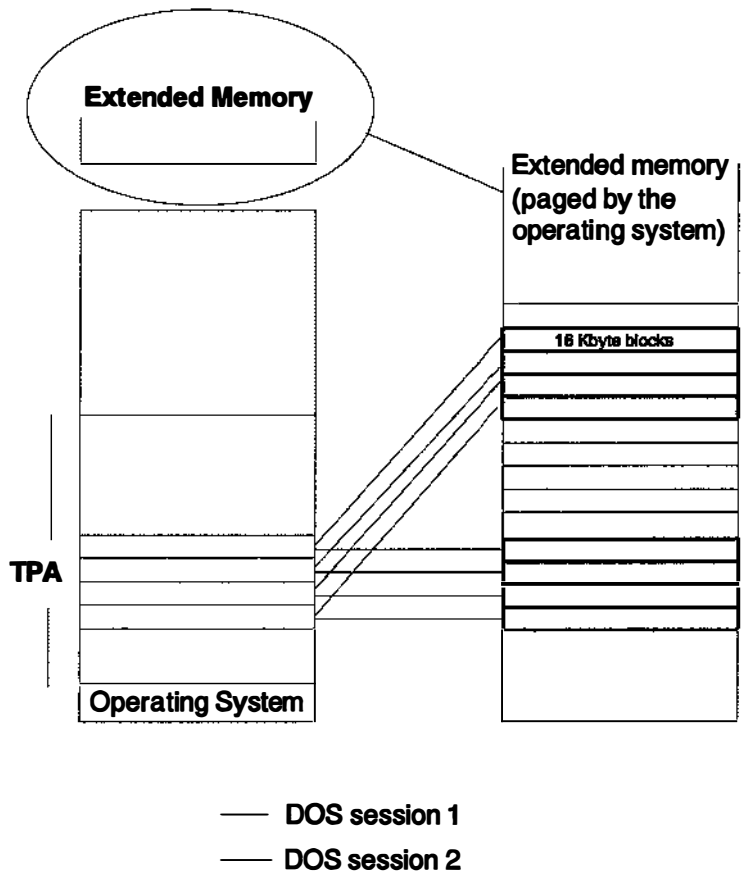


Figure D-4 Paging from Extended Memory

The operating system uses paging to perform multitasking. Programs are allocated blocks of physical (extended) memory. When it is the turn of one program to run, the blocks used by that program are mapped into the TPA and appear to be in conventional memory, that is they have a logical address below the 640 Kbyte barrier and the program is said to be *in context*. After one time slice another program is paged into the TPA and the original program joins the queue waiting for its next turn to be brought into context.

A paging table is set up which holds a record of all the memory in the system and keeps track of which areas are used for what. All memory can then be paged into conventional memory.

LIM (expanded) memory

A special use of memory is the use of the *LIM EMS (Expanded Memory Specification) standard*. Devised by Lotus, Intel, and Microsoft, LIM allows application writers to access memory above the 640 Kbyte conventional memory barrier *but only for data*. This feature is useful when you are running applications with large databases or spreadsheets such as AutoCAD® and Lotus 1-2-3 (Version 2.2). However, it requires that programs have been designed specifically to access it.

Your system can be configured to provide LIM emulation memory in one of a number of ways using the INSTALL or SETUP programs.

Page frame

In the first, “traditional” method, a 64 Kbyte page frame is created in upper memory during initialization into which four 16 Kbyte blocks of expanded memory can be paged. This effectively increases the TPA; the program code is paged into the TPA, some or all of the data (that would otherwise also have needed to be in the TPA) is paged into the LIM page frame. Different blocks of data can be paged in and out of the page frame as necessary while the program is running.

The drawback of using this method of LIM emulation is that the 64 Kbyte page frame resides permanently in upper memory using blocks that would otherwise have been used for the operating system. This means that more conventional memory is used for the operating system, reducing the TPA for *all* applications.

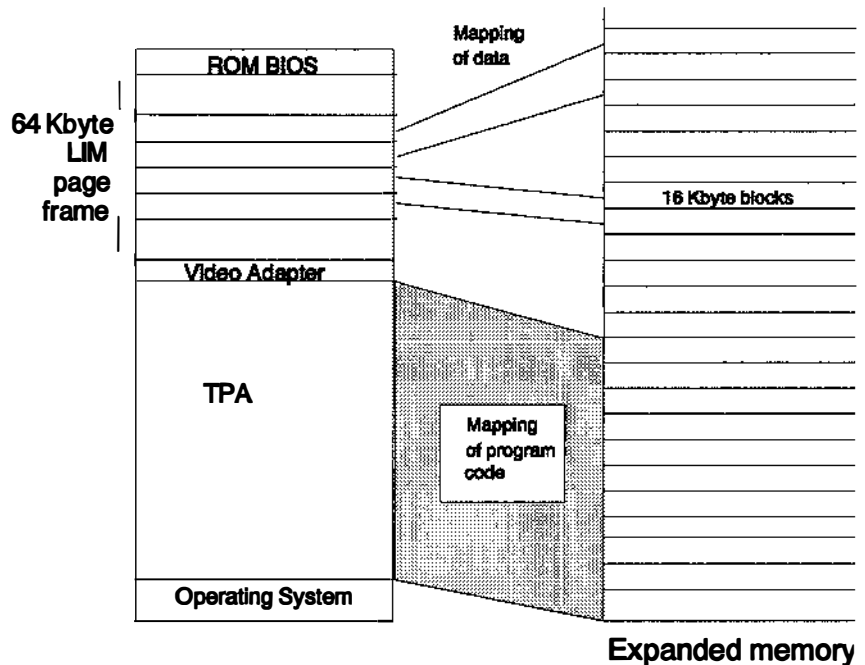


Figure D-5 Paging with LIM Emulation

An alternative method, which overcomes this limitation, sets the page frame up in the TPA. Programs that do not use LIM will then have the same amount of memory as if LIM were not enabled. When a program running LIM is started, 64 Kbytes are set aside for LIM paging, the rest is used for program code in the normal way. (Note that the program must first have been set up to use LIM through the PIFED command.)

This option is also useful if you do not have room in upper memory for the 64 Kbyte page frame, for example if you have several extra network cards or disk controllers fitted.

LIMSIZE command

The amount of expanded memory allocated to each program using LIM for the pages to be paged into the LIM page frame is set by the LIMSIZE option. It is important to set this option realistically, so that they do not use all the expanded memory available, preventing other applications from running.

Note that this is in addition to the memory allocated for the program code which is determined by the `INSTALL` or `SETUP` option `Max memory per process` or by the `MEMSIZE` command.

Non-banked memory

You can run programs in “non-banked” memory. This means that they occupy part of the TPA all the time they are running, reducing the area available for paging other applications (sometimes called the *banked window*). However, this may be useful if several programs need to share memory (usually data). A DOS session providing the common memory is run in non-banked memory so that it is always in context.

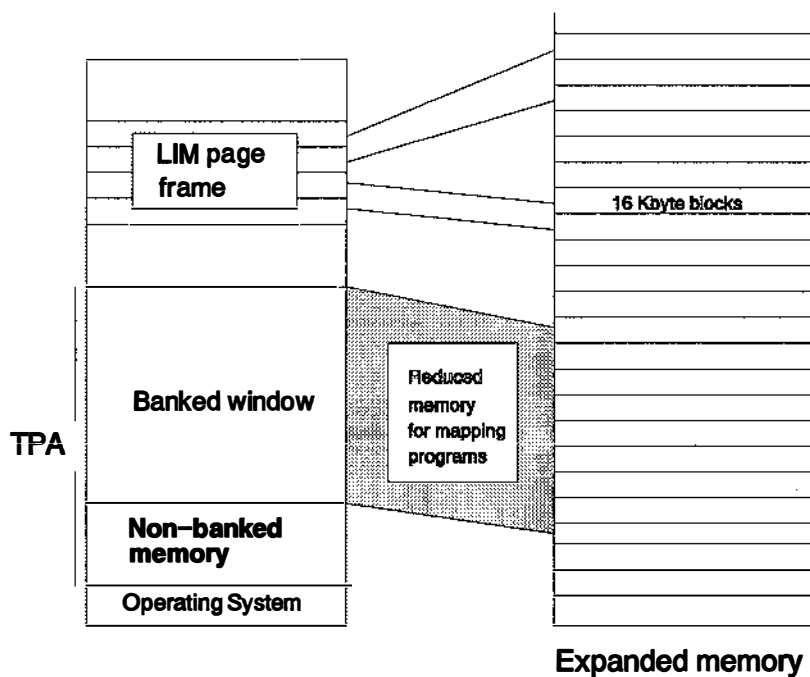


Figure D-6 Using Non-banked Memory

Checking on your computer's memory usage

To find out what memory you have on your computer, and how it is configured, use the `SHOW /M` command.

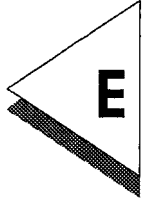
Commands that control memory usage

The operating system provides a number of commands that you can use to tailor the way memory is used. They supplement the configuration you do using `INSTALL` and `SETUP`. For more information about these commands see Chapters 11, "Commands," and 16, "Customizing your system."

The commands are:

- ☐ `ALIGN` - determines whether the operating system loads a program in aligned memory.
- ☐ `LIMSIZE` - sets the maximum amount of expanded (paged) memory available to an application.
- ☐ `MEMSIZE` - sets the maximum amount of application memory available to an application.
- ☐ `PIFED` - associates system parameters with an application. `PIFED` overrides some values set with other commands.

Using a setup input file



THIS appendix explains how to place system configuration values directly in a setup input file, and so customize a system without having to run `SETUP` or `INSTALL` as described in Chapters 1, “Doing an installation,” and 16, “Customizing your system.”

Note that this method is more open to error than the standard approach, and is only recommended where minimum changes are being made to an existing configuration, or where the correctness of the file has already been verified.

E.1 The setup input file

Whenever **SETUP** or **INSTALL** is run, an ASCII file is created, containing the system configuration settings in human readable form. This setup input file has the same name as the system file identified in **SETUP** or **INSTALL**, but with the **.inp** extension.

If you wish to create a new configuration, you can, instead of rerunning **SETUP** or **INSTALL** and working your way through all the displays these utilities offer you, edit the **.inp** file directly, or create a new setup input file altogether. You can then specify the new or modified **.inp** file in the following command:

```
SETUP /I systemfilename [inputfilename]
```

This writes the configuration values given in the file identified by *inputfilename* to the system file given in *systemfilename*, overwriting the previous settings.

For example, to modify a system file called **os.sys** by including values from a setup input file called **newsys.inp**, you would type:

```
C>SETUP /I os.sys newsys.inp
```

If no input file is specified, a default of system file name plus **.inp** extension is assumed.

Any values not explicitly set in a **.inp** file remain unchanged in the resulting system file.

E.2 Input file values

A setup input file contains a series of statements, each assigning particular values to setup fields. Statements take the form:

```
category [device...] keyword=value...
```

Here *category* represents one of the following five classes of statement:

Category	Topic
SETMEM	Memory setup
SETMISC	Miscellaneous items
SETMULTI	Multiport parameters
SETPORT	Serial port parameters
SETTERM	Terminal parameters

The *device* parameter is only applicable to the categories SETPORT and SETTERM.

In the case of SETPORT, *device* indicates the particular port to be configured. With SETTERM, *device* indicates the particular terminal (or “station”) to be configured.

Note

SETPORT statements should always precede SETTERM statements, so that terminal or station numbers can be matched to ports.

The *keyword=value* parameters vary from category to category, and are described in the sections that follow.

Input file statements can be entered in uppercase, lowercase or a combination of the two.

Statements, and the settings they contain, can be entered in any order in a setup input file (with the proviso that SETPORT statements must precede SETTERM statements).

Numeric values can be entered in decimal or hexadecimal.

Separate each *keyword=value* combination from the next by a comma, a space or a new line.

SETMEM statements

SETMEM statements let you establish memory management settings. They have the format:

SETMEM keyword=value...

where valid keywords and values are as follows:

Table E-1 SETMEM Settings

Keyword	Meaning	Valid values
HIMEMAUTO	Autoscan high memory (one bit to be set for each 16K block from C0000 to F4000)	0 to 0xffff
HIMEMUSE	Enable/disable high memory (as above)	0 to 0xffff
LIM	Enable/disable LIM	Yes (enabled) /no (disabled)
LIMBASE	LIM page frame base address	0 = TPA 0xffff = Auto C000-F400 = specified address
LIMMAX	LIM max 16K pages per process	0 to 0xffff (16K pages)
PROCMAX	Maximum memory per process	0 to 0xffff (paras)
SYSDAT	Reserve sysdat space	0 to 0xffff

SETMISC statements

SETMISC statements let you establish a number of miscellaneous settings. They have the format:

```
SETMISC keyword=value...
```

where valid keywords and values are as follows:

Table E-2 SETMISC Settings

Keyword	Meaning	Valid values
MGDISK	Multigraphics disk buffers	0 to 8
MGMEM	Multigraphics memory buffers	0 to 8
RESET	Enable/disable Ctrl+Alt+Del global reset	Yes (enabled) /no (disabled)
SCROLL	Color monitor scroll mode	Fast/slow/wideopen
SWITCHMENU	DOS session switch keys, to work with Esc to display menu	Any combination of Ctrl , Shift and Alt separated by plus signs
SWITCHNEXT	DOS session switch keys, to work with Esc to display next DOS session	Any combination of Ctrl , Shift and Alt separated by plus signs
SWITCHNUM	DOS session switch keys, to work with numeric keypad to display specified session	Any combination of Ctrl , Shift and Alt , separated by plus signs
SYSHANDLES	Maximum number of open files	0 to 0xffff
TIMER	Timer 0 register value	0 to 0xffff
VCHANDLES	Maximum number of open files per process	0 to 0xffff
VERIFY	Diskette verify after write	Yes/no

SETMULTI statements

SETMULTI statements let you establish settings for a multiport board. They have the format:

SETMULTI keyword=value...

where valid keywords and values are as follows:

Table E-3 SETMULTI Settings

Keyword	Meaning	Valid values
BASE	Base port address to check	0 to 0xffff (0 = no board)
IRQLATCHP	IRQ latch port	0 to 0xffff (0 = none)
IRQMASK	Mask and polarity or IRQ latch bits	0 to 0xffff
IRQNUM	IRQ number	0 to 0xffff
NPORTS	Number of ports on card	0 to 64
RESDAT	Reset r/w flag and bit pattern	0 to 0xffff
RESPORT	Reset port	0 to 0xffff (0 = none)

SETPORT statements

SETPORT statements let you establish settings for serial ports. They have the format:

```
SETPORT [device...] keyword=value...
```

Here, *device* indicates the particular port to be configured, and takes the forms:

P_n

P_n, P_n,...

P_n-P_n

to indicate a single port or series of ports, where *n* is a number between 0 and 64.

Valid keywords and values are as follows:

Table E-4 SETPORT Settings

Keyword	Meaning	Valid values
BAUD	Baud rate	50 - 115000 (to nearest integer)
PARITY	Parity	None, odd, even
PROT	Protocol	None, or any combination of DTR, RTS and XON, separated by plus signs
STOPBITS	Stop bits	1, 2
TYPE	Port function	Printer, term, mouse
WORDLEN	Word length	7, 8

SETTERM statements

SETTERM statements let you establish settings for terminals. They have the format:

SETTERM [device...] keyword=value...

Here, *device* indicates the particular terminal (or “station”) to be configured, and takes the forms:

Sn

Sn, Sn,...

Sn-Sn

to indicate a single terminal or series of terminals, where *n* is a number between 0 and 63.

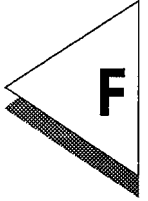
Instead of S0, you can enter MAIN to indicate the main console.

Valid keywords are as follows:

Table E-5 SETTERM Settings

Keyword	Meaning	Valid values
KEYBN	US/national keyboard	US, nat
KEYS	84 char/enhanced keyboard	84, 102
MOUSE	Mouse port/no mouse	0 to 64, or 0xff for no mouse (the specified port should be set to type=mouse)
NVCS	Number of DOS sessions	1 to 8
PCTERM	PC terminal/dumb terminal	Yes (for PC terminal), no (for dumb terminal)
STATLINE	Enable/disable status line	Yes (enabled), no (disabled)

Escape code sequences



ANSI and VT52 escape code handling is built into the operating system. Escape code sequences provide extra options for programs that need to move the cursor, alter the screen display, or assign keyboard use.

F.1 ANSI escape sequences

The general form of an ANSI escape sequence is:

- ☐ A one byte ESC character (hexadecimal 1BH).
- ☐ A one byte [character (hexadecimal 5BH).
- ☐ One or more decimal numbers separated by semicolons.
- ☐ A single command character.

There follows a list of the ANSI escape sequences. Note:

- 1

 y, x indicate row and column numbers, which range between 1 and 50 rows (with 1 at the top of the screen) and 1 and 80 columns (1 at the left of the screen). The numbers are specified in ASCII digits. If you omit the value (or specify zero), the default value is used.
- 2

 Spaces are shown only to make the sequences clearer to read.

The sequences are divided into the following types:

- ☐ Cursor positioning.
- ☐ Erasing.
- ☐ Inserting.
- ☐ Changing display and screen modes.

Cursor positioning

Cursor position (CUP) ESC [$y;x$ H

Horizontal and vertical position (HVP) ESC [$y;x$ f

CUP and HVP position the cursor according to the co-ordinates issued. The default (and null) value is the top left corner of the screen. CUP and HVP are equivalent.

Cursor up (CUU) ESC [y A

Moves the cursor up y rows without changing columns. The sequence is ignored if the cursor is already at the top line.

Cursor down (CUD) ESC [y B

Moves the cursor down y rows without changing columns. The sequence is ignored if the cursor is already at the bottom line.

Cursor forward (CUF) ESC [x C

Moves the cursor to the right without changing rows. The sequence is ignored if the cursor is already in the far right column.

Cursor backward (CUB) ESC [x D

Moves the cursor to the left without changing rows. The sequence is ignored if the cursor is already in the far left column.

Device status report (DSR) ESC [6 n

The DSR sequence instructs the console driver to return a sequence that reports the cursor position in terms of row and column. It is returned via the keyboard input.

Save cursor position (SCP) ESC [s**Restore cursor position (RCP)** ESC [u

SCP stores the current cursor position. RCP restores the cursor to that position. The default for RCP, if no SCP was issued, is 0,0. These escape sequences cannot be nested.

Erasing

Erase display (ED) ESC [2 J

Erases the display and moves the cursor to the top left of the screen.

ESC [J or ESC [0 J erases from the cursor to the end of the display. ESC [1 J erases from the beginning of the display to the cursor.

Erase line (EL) ESC [K or ESC [0 K

Erases at the cursor position and to the end of the line.

ESC [1 K erases from the beginning of the line to the cursor.

ESC [2 K erases the entire line.

Delete line (DL) ESC [M

Deletes line.

Inserting a line

Insert line (IL) ESC [L

Inserts line.

Changing the display and screen modes

Set graphics rendition (SGR) ESC [*n* ; ... ; *n* m

Sets various display modes which remain until a different SGR sequence is set; display modes are shown in Table F-1.

Table F-1 ANSI Display Modes

Value for <i>n</i>	Description
0	Return to default (white characters on black background)
1	High intensity (bold)
4	Underscore (monochrome only)
5	Flashing
7	Inverse video (black characters on white background)
8	Invisible (black on black)
30	Black foreground
31	Red foreground
32	Green foreground
33	Yellow foreground
34	Blue foreground
35	Magenta foreground
36	Cyan foreground
37	White foreground
40	Black background
41	Red background
42	Green background
43	Yellow background
44	Blue background
45	Magenta background
46	Cyan background
47	White background

Set screen mode (SM) ESC [= n h 1 or ESC [= h 1 or ESC [
? n h

Reset screen mode (RM) ESC [= n 1 1 or ESC [= 1 1 or ESC
[? n 1

These sequences set and reset the screen operating mode. *n* may take the values shown in Table F-2.

Table F-2 ANSI Screen Modes

Value for <i>n</i>	Description
0	40 x 25 monochrome
1	40 x 25 color
2	80 x 25 monochrome
3	80 x 25 color
4	320 x 200 color
5	320 x 200 color
6	640 x 200 monochrome
7	Enables (SM) or disables (RM) wrapping at the row end. When this parameter is enabled, extra characters wrap on to the next line. When disabled, they are lost.

You can use extended character codes (ECCs) by typing the first variable in the sequence as 0 (NUL) followed by the second key. For example, 0;68; is the F10 key. The last variables are the character codes for the values you are assigning to the key. An alternative to the sequence of numbers is a string defined as characters enclosed in double quotation marks. For example, 65;66;67 can be shown as "ABC";.

Examples

To position cursor at row 10, column 10, type:

ESC [10;10H

To set screen mode to 40x25 color, type:

ESC [?1h

To reset current attributes and then set green as foreground color on a white background, type:

ESC [0;32;47m

F.2 VT52 Escape sequences

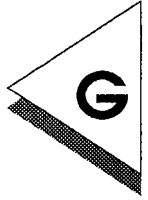
The VT52 escape sequences are listed in the following table.

Table F-3 VT52 Escape Sequences

Parameter	Description
ESC A	Cursor up
ESC B	Cursor down
ESC C	Cursor forward
ESC D	Cursor backward
ESC E	Erase console
ESC H	Home cursor
ESC I	Reverse index
ESC J	Erase to end of screen
ESC K	Erase to end of line
ESC L	Insert a blank line
ESC d	Erase from beginning of screen
ESC l	Erase entire line
ESC o	Erase from beginning of line
ESC M	Delete one line
ESC N	Delete one character
ESC Y	Set cursor position
ESC b	Set foreground color
ESC c	Set background color
ESC e	Enable cursor
ESC f	Disable cursor
ESC j	Save cursor position
ESC k	Restore cursor position
ESC p	Reverse video on
ESC q	Reverse video off

Table F-3 VT52 Escape Sequences (continued)

Parameter	Description
ESC r	Intensity on
ESC u	Intensity off
ESC s	Blink on
ESC t	Blink off
ESC v	Wrap at line end on
ESC w	No wrap at line end
ESC x	Switch console to color monitor
ESC y	Switch console to color monitor
ESC z	Restore normal attributes
ESC a	Set color graphics card video mode
ESC 0	Turn status line off
ESC 1	Turn on monochrome status only
ESC 2	Turn on color status only
ESC 3	Turn on both status lines
ESC 4	Turn off status line clock display
ESC 5	Enable status line clock display
ESC 6	Turn off pfk expansion
ESC 7	Turn on pfk expansion



Sample menuing batch file

THIS appendix shows a sample batch file that collects certain common commands, together with the batch files offered as examples in Chapter 15, "Running applications," and arranges them as a menu of options on the screen.

Using batch files similar to the one shown, Superusers can create personalized user interfaces for the accounts on their systems.

```

MENU

X Leave MENU system

L Lock Terminal
U Show Users
m Check Mail
M Mail
Q Logout

0 LetterPerfect
1 Lotus 123 Release 2.2
2 Lotus Manuscript
3 DataEase 4.2
4 Paradox v3.5
5 SuperCalc 5.0D
6 Microsoft Word 5.0
7 WordPerfect 5.1
8 Q&A v3.0 LAN

Choose option (Type one character)

```

```
IF %choice% == 4 CALL paradox.bat
IF %choice% == 5 CALL sc5.bat
IF %choice% == 6 CALL word.bat
IF %choice% == 7 CALL wp.bat
IF %choice% == 8 CALL q&a.bat
GOTO menu
:mail
ECHO
ECHO
ECHO
ECHO
ECHO
INPUT mailaddress
MAIL %mailaddress%
GOTO menu
:users
SHOW /u
PAUSE
GOTO menu
:end
SET choice=
```

Type user name of recipient and press Enter To look at your mail just press Enter
--

Glossary



THIS appendix gives a brief explanation of terms and concepts used in this manual that may be unfamiliar to you.

Access permissions

The kind of file or directory access allowed to the various classes of user, *owner*, *group* and *world*. Access permissions can be any combination of read, write and delete.

Application

A piece of software used to carry out a particular task, like a word processor or graphics package.

ASCII

The American Standard Code for Information Interchange is a standard code for representing numbers, letters and symbols. An ASCII text file is a file that can be intelligibly displayed on the video screen or printed on paper.

Autoexec.bat file

This is a configuration file which the operating system executes every time it is started. The easiest way to change the **autoexec.bat** file is to use the **SETUP** program.

Backup

A copy of a file, directory or disk made for safe keeping.

Batch file

A file containing a sequence of commands. When you type the name of the file at the command line, each command in the file is carried out in turn, just as if you had typed the commands individually. The filename becomes in effect a “command” in itself. Batch filenames can be up to eight characters in length, but must always have the extension **.bat**.

Baud rate

The baud rate is the number of bits transmitted per second in serial communications.

Bit

A bit is the smallest unit of memory in a computer. Bits are electronic switches. They are given values to indicate whether they are switched on or off, 0 is off and 1 is on. Eight bits are equal to one byte, or character. See also *Byte*.

Buffer

An area of memory that temporarily stores data during information transfer.

Built-in command

See *Internal* command.

Byte

A unit of memory or disk storage usually used to represent one character. See also *Bit*.

Cache

A cache or disk cache is an area of RAM memory used to store data accessed frequently from disk. If your applications want to access a disk for data, it can become time-consuming. To speed up the process of reading and writing data to and from disk, you can set up a cache. Whenever data is read or written from disk, a check is made to see if a copy is in the cache first. If it is in the cache, the copy is used. Caches are very useful for increasing the performance of applications that have to access a lot of data from disk, like databases and spreadsheets.

Cconfig.sys file

A configuration file which the operating system executes every time it is started. It contains various commands used to set up your system for items like keyboard and country information. The easiest way to change the **cconfig.sys** file is to use the SETUP program.

Command

A short operating system program designed for a specific task.

Command line

The command followed by any other relevant information given at one time to the computer, to enable it to carry out your instructions. The command line ends when you press the **Enter** key.

Communications port

A socket to which another computer may be attached usually via a cable, thus allowing the computers to communicate with each other.

Conventional memory

Memory below 640 Kbytes, sometimes called base memory. It is generally where the operating system and applications are loaded. It is usually referred to as RAM (Random Access Memory).

Current disk drive

The drive on which the operating system first looks for files. The current drive letter is usually displayed in the system prompt.

Cursor

A blinking line or block on the screen that indicates where your next keystroke will have an effect.

Databits

Refers to the number of bits used to define a character on serial data transmission, usually 7 or 8. When connecting two computers together via the communication ports, it is important to set the same number at each end.

Default

A pre-set value used by a computer until it is specifically changed by a user.

Device

A device is a piece of hardware attached to a computer (usually a peripheral), for example a printer or mouse.

Device driver

Device drivers are programs that are loaded alongside the operating system to operate hardware devices which are not recognized as a default part of the system. Device drivers usually have the extension **.sys**.

Directory

A place on disk where filenames are stored so that the operating system can find the files when needed. Each disk can contain many directories.

Disk

Magnetic medium used for storing information.

Disk cache

See *Cache*.

Diskette

A disk that can be removed from the drive (also known as floppy disks).

Disk operating system

A collection of programs that manages computer resources and other programs on a computer.

DOS session

One of a number of separate working environments on a multitasking terminal, similar to the different channels on a television. Each DOS session can run its own applications independently of the others.

Drive

The device that holds and reads a hard disk or diskette.

EDITOR

A program that allows you to create and edit ASCII text files on your system.

EMM

Expanded Memory Manager.

EMS

Expanded Memory Specification.

Enter key

The key that tells the operating system to carry out a command. Also referred to as **Return** or ↵.

Executable

This means “programs that can be run by the computer.” Executable code is a series of instructions that can be carried out by the computer. Executable programs usually have the extension **.exe** or **.com**.

Expanded memory

Expanded memory is often known as LIM memory. The LIM EMS standard was jointly devised by Lotus, Intel and Microsoft®. EMS is the Expanded Memory Specification. It describes ways in which a program may access up to 32 Mbytes of memory outside conventional memory. Programs that use expanded memory include spreadsheets, databases, and other programs that need access to large amounts of data.

Extended memory

Memory above 1 Mbyte on 80386- and 80486-based computers.

External command

Commands that are used less often than other commands, and so are not automatically loaded into memory. To save space they are stored on disk instead. They have the file extension **.com** or **.exe**. When you enter an external command, the operating system retrieves the “command file” from disk and runs it. The memory used by the command is freed when it has finished its work. See also *Internal command*.

File

A collection of related instructions or data stored on disk.

File Allocation Table (FAT)

The File Allocation Table is a kind of index which exists at the start of every disk, and is used by the operating system to locate entries for files.

Filename and filename extension

The name assigned to a file. A filename can include a primary filename of 1 through 8 characters and optional filename extension of 1 through 3 characters. A period (.) separates the filename from the filename extension.

Filespec

Short for file specification. The combination of drive letter, path, and filename that identifies a file uniquely to the operating system.

Format

The process by which the operating system prepares a disk for use by writing electronic markers onto the disk, so that it can store and retrieve files. FDISK is used to format hard disks and the FORMAT command is used to format diskettes. When you format a disk, all the information previously stored on the disk is erased, so take care when you use these commands.

Group

The collection of related users to which a particular user on a “secure” system belongs.

Hard disk

A disk for storing information that is not removable from the computer. They usually have large storage capacity and provide fast access to data.

High memory

The first 64 Kbytes of extended memory.

Input

Information going into the computer, usually from you typing at the keyboard or from a program reading from disk.

I/O

Abbreviation for input/output.

Internal command

An operating system command that usually resides in memory. These commands respond quickly because they do not have to be loaded from disk.

Kilobyte (Kbyte or KB)

1024 bytes, denoted as 1 Kbyte, 1 KB or 1K. 32 Kilobytes equal 32 Kbytes. 1024 Kbytes equal 1 Megabyte (or Mbyte), over one million bytes.

Label

A marker within a batch or cconfig.sys file. It is used to indicate the specific point that execution jumps to, following a GOTO statement.

LIM

See *Expanded memory*.

Logical drive

The operating system's internal representation of a drive. It may refer to an actual disk device, or to a group of directories specified using the SUBST command.

Megabyte (Mbyte or MB)

A unit of storage in a computer denoting 1024 Kbytes. It is usually denoted as 1 Mbyte or 1 MB.

Memory

Memory is where the computer stores data and programs. It is measured in bytes. One byte is equal to a group of eight bits. A bit is the smallest unit of memory in a computer. Each bit has the value 0 or 1. Memory is thought of in bytes because one byte of memory can be used to represent a character such as "a," "?," or "4." One thousand and twenty-four bytes is equal to one *Kilobyte* or (*Kb*) for short. As computers contain a lot of memory, memory is usually talked about in terms of Kilobytes rather than bytes.

Memory disk

See *Virtual disk*.

Modem

Modulator/Demodulator. A device which converts a digital signal from a computer into an analogue signal which can then be transmitted down telephone lines and decoded by a modem at the receiving end.

Multitasking

The term used to describe the operating system's capacity to perform multiple operations simultaneously.

Multiuser

The term used to describe the operating system's capacity to support more than one terminal, and so more than one user, at the same time.

Online

A computer or peripheral is “online” when it is switched on and ready to accept information.

Operating system

See *Disk operating system*.

Output

Data that the computer sends to the console, disk or some other device.

Owner

The user in a “secure” system to whom a particular file or directory belongs.

Parallel

The carrying out of several tasks at the same time. For example in parallel communications, 8 bits of a byte would be transmitted together (see also *Serial*).

Parity

A means of checking that data has been transferred correctly, either between computers across a communications link, or between components inside a single computer, such as a disk and memory.

Partition

A partition is a section of a hard disk created by the FDISK disk preparation utility.

Password

A series of characters needed to gain access to a computer, file or directory.

Path

The description of the route through the directory hierarchy to a subdirectory or file. A drive letter can also be included at the beginning of the path.

PC

An abbreviation for personal computer.

Peripheral

An external device connected to the computer. Peripherals are generally used for input and output. Examples include disks, modems, and printers.

Piping

The process where the output from one program is made to be the input for another.

Program

A series of specially coded instructions that perform a specific task when executed by a computer.

RAM

Random Access Memory is a type of memory that is used to load and run programs from. It can be thought of as the computer's workspace. The more complex a task you want to do, the more RAM memory your computer needs. It is memory that can be read to, as well as written from, in any order. See also *ROM*.

RAMdisk

See *Virtual disk*.

Read-only (RO)

An attribute that can be assigned to a file or directory. When switched on, the read-only attribute allows you to read from the file but not to write any changes to it.

Real time

A system where execution depends on critical timing criteria. For example a system may be required to respond to some event within a given time.

RETURN

See **Enter** key.

ROM

Read-Only Memory. This is a type of memory which you can only read or copy; you cannot alter its contents. It is normally used to contain the computer's diagnostic programs, and a very basic input/output system (or BIOS).

Root directory

The main directory that the operating system creates on disk when the disk is formatted.

Scrolling

Scrolling means to move text up or down on a screen so that you can read it all.

Serial

The carrying out of tasks one after the other. For example, in serial communications, each byte is transferred one bit at a time (see also *Parallel*).

SID debugging tool

A powerful debugger which allows you to test and debug programs interactively.

Stopbits

A term used in serial data transfer to distinguish where one character starts and another stops, communication programs normally add one, or sometimes two, stopbits on the end of each character. Like databits, it is important that the number of stopbits set up agrees at both ends of the line.

String

A sequence of characters, such as “hello” or “arc132.”

Subdirectory

Any directory that is not the root directory on a disk can be called a subdirectory.

Supergroup

The group to which the *Superuser* belongs. Additional users can be assigned to this group. Supergroup members other than the Superuser have access rights equivalent to those of the owner’s group for each file and directory in the system.

Superuser

The person given responsibility for installing and managing a multiuser system. On “secure” systems with user accounts established, the Superuser has access rights equivalent to those of the owner for each file and directory in the system.

System prompt

The display on the screen that shows you the operating system is ready to receive a command. The standard system prompt can be changed by using the PROMPT command.

Timeout

The operating system sets a time limit for response to a command, so it can detect if something is taking too long to respond (for example a printer may have inadvertently been switched off), and return an error message.

Upper memory

Memory between 640 Kbytes and 1 Mbyte (384 Kbytes) reserved for use by the video display adapter, add-on adaptors, and system software.

Utility

An operating system program that enables you to perform certain operations such as copying, erasing, or editing files.

Variable

In batch files, a variable is a character that is substituted for a filename when the batch file is run.

Video memory

A special area of memory used by the hardware which operates the computer's display or monitor.

Virtual disk

A portion of computer memory that is made to act like a very fast disk. You store your files on it in the same way as on a physical disk. Information stored on a virtual disk is lost when the computer is switched off or when the operating system is restarted. Virtual disks are therefore best used for temporary storage of expendable data. They are also referred to as *RAM disks* or *memory disks*.

Volume label

The name given to a disk for identification. It is set by the LABEL command. You can display the volume label on a disk using the VOL command.

Wildcard characters

Special characters that match certain specified items. There are two wildcard characters, ? and *. The ? can be substituted for any single character in a filename and the * can be substituted for one or more characters in the filename or filename extension, or both.

Wildspec

The same as *filespec* except that you can use wildcard characters in the command syntax to specify groups of files.

World

The remaining users on a “secure” system that fall into neither owner nor group categories.

Write-protected

A write-protected disk, directory or file can be read but not altered or erased. Disks are write-protected physically, usually by the absence or presence of a notch on the diskette casing. Files are write-protected via the ATTRIB command (setting the read-only attribute), and files and directories can be write-protected with the PASSWORD command.

Index

- % in batch file processing, 10-4
- %% variable, FOR command, 10-14
- * wildcard, 7-4
- .bak files (EDITOR), 9-12
- .bat files, 10-1
- .com files, 6-2
- .exe files, 6-2
 - converting to .bin or .com, 11-56
- /H help switch, 11-2
- 3 1/2 inch diskettes, 5-3
- 5 1/4 inch diskettes, 5-3
- 8086 mnemonics, assembling into memory, 14-8
- < redirection symbol, 12-2
- > redirection symbol, 12-2
- ?
 - conditional execution of cconfig.sys commands, 16-6
 - SID command, 14-7
 - wildcard, 7-4
- ?, cconfig.sys command, 16-6
- ?? (SID command), 14-7
- @, syntax for filelist, 7-5, 11-7
- @command batch subcommand, 10-9
- :label batch subcommand, 10-10
- | piping symbol, 12-4

A

- access permissions, 2-2, 8-3
 - default, changing, 8-6
 - default, displaying, 8-6

- specific, changing, 8-5, 11-150
 - specific, displaying, 8-5, 11-150
- access permissions, definition of, H-2
- address, 20-bit specified in SID, 14-3
- ALIGN command, cconfig.sys file, 16-8
- aligned memory, use by block device drivers, 16-8
- aligning
 - display on a CGA monitor, 11-98
- aligning display on CGA monitor, 11-98
 - escape code sequences, F-2
 - extended character codes, F-5
- APPEND command, 11-11
- applications
 - amending settings for, 15-8
 - configuring for the spooler, 13-14
 - copy-protected, 15-20
 - improving performance, 15-1
 - multiple copies, running, 15-18
 - running, 15-1
 - starting from autoexec.bat, 16-38
 - TSRs, 15-17
- archive attribute, 8-12, 11-16
- ASCII, definition of, H-2
- ASSEMBLE (SID command), 14-8
- assembly language syntax, A and L commands, 14-38
- ASSIGN command, 11-14 - 11-15
- assigning
 - drive letters, 1-13, 16-24
 - passwords, 8-9, 11-105
 - printers (MODE), 11-100, 13-19
- ATTRIB command, 11-16 - 11-17
- attributes, file, 11-16

- autoexec.bat file, 16-36
 - definition of, H-2
 - SETUP, use by, 16-3
 - starting applications from, 16-38
- automatic backup files (EDITOR), 9-12
- auxiliary device, listing, 11-116
- auxiliary ports, 1-14, 11-115, 13-2, 16-42
 - see also serial ports*
- AUXPORT command, 11-18

B

- backing up
 - definition of, H-2
 - directories, 8-15 - 8-16
 - hard disks, 8-13
 - modified files, 8-15
- backup
 - disks, checking contents of, 8-17
 - files, EDITOR, 9-12
 - log, 11-20
- BACKUP command, 8-13, 11-19 - 11-22
 - error codes, 11-21
- BANK command, 11-23, 15-6
- banked memory, D-7
- base memory, D-2
- basic concepts, 5-2
- batch file processing, 10-1
- batch files, 10-1
 - creating, 10-3
 - definition of, H-2
 - environment variables, 10-6
 - general purpose, 10-4
 - running, 10-3
 - running applications from, 15-21
 - sample menuing, G-1, G-3
 - stopping, 10-4
- batch subcommands
 - conventions, 10-8
 - @command, 10-9
 - :label, 10-10
 - CALL, 10-11
 - definition of, 6-4
 - ECHO, 10-12
 - ECHOERR, 10-13
 - FOR, 10-14
 - GOTO, 10-15
 - IF, 10-16
 - INPUT, 10-19
 - INPUTC, 10-20
 - PAUSE, 10-21
 - PAUSEERR, 10-22
 - REM, 10-23
 - SHIFT, 10-24
 - summary table, 10-6
- baud rates
 - definition of, H-2
 - FILELINK command, 11-71
 - MODE command, 11-102, 11-104, 13-18
- bit, definition of, H-3
- BLOCK COMPARE (SID command), 14-9
- blocks of text, in EDITOR, 9-10
- bootable partitions, selecting, 11-66
- BREAK command, 11-24
 - cconfig.sys file, 16-10
 - command line, 11-24
 - setting on or off, 16-10
- breakpoints, 14-13, 14-18
- buffers
 - definition of, H-3
 - setting number of, 16-11
- BUFFERS, cconfig.sys command, 16-11
- buffers, setting number of, 1-12
- built-in commands
 - see internal commands*
- byte, definition of, H-3

C

- cache, 16-30
 - setting up in cconfig.sys, 16-30
 - configuring in INSTALL or SETUP, 1-12
 - definition of, H-3
 - setting up at the command line, 11-25, 16-30
- CACHE command, 11-25
 - cconfig.sys file, 16-30
- CALL batch subcommand, 10-11
- cconfig.sys, 16-4
 - command summary, 16-5
 - definition of, H-3

- DOS applications, use by, 15-17
- SETUP, use by, 16-3
- cconfig.sys commands
 - ?, 16-6
 - ALIGN, 16-8
 - BREAK, 16-10
 - BUFFERS, 16-11
 - COUNTRY, 16-12
 - definition of, 6-4
 - DEVICE, 16-14
 - ECHO, 16-16
 - ENVSIZE, 16-17
 - FASTOPEN, 16-18
 - FILES, 16-19
 - FLAGS, 16-20
 - GOTO, 16-21
 - IDLE, 11-83, 16-22
 - INSTALL, 16-23
 - LASTDRIVE, 16-24
 - NETWORK, 16-25
 - QUEUES, 16-26
 - REM, 16-27
 - SET, 16-28
- CD command, 7-13, 11-26 - 11-27
- CGA monitor
 - aligning display (MODE), 11-98
- characters
 - not allowed in disk labels, 11-88
- CHDIR command, 7-13, 11-26 - 11-27
 - special characters, 11-26
- checking
 - contents of backup disks, 8-17
 - data is correctly written to disk, 11-148
 - disks, 11-28
- checklist, troubleshooting, A-44
- CHKDSK command, 11-28 - 11-31
- classes of user, 8-3
- CLEAR_SCREEN, in config.cpl, 13-14
- clearing
 - passpoints, 14-18
 - screen display, 11-32
- CLS command, 11-32
- clusters, lost, 11-28
- columns, setting display number of, 11-99
- combining
 - files, 11-39
 - pipes and redirection, 12-6
- COMMAND command, 11-33 - 11-34
- command line
 - definition of, 6-1, H-4
 - editing, 11-9 - 11-10
 - getting started at, 5-10
- command prompt
 - see system prompt*
- command switches, definition of, 6-5
- command syntax, 11-7
- command, definition of, H-3
- command.com, 11-33, 15-5
- commands, 11-1
 - ? (cconfig.sys), 16-6
 - ALIGN (cconfig.sys), 16-8
 - APPEND, 11-12
 - ASSIGN, 11-14
 - ATTRIB, 11-16
 - autoexec.bat, 16-36
 - BACKUP, 11-19
 - BANK, 11-23, 15-6
 - batch subcommands, 6-4, 10-6
 - BREAK, 11-24
 - BREAK (cconfig.sys), 16-10
 - BUFFERS (cconfig.sys), 16-11
 - CACHE, 11-25
 - cconfig.sys file, 16-4
 - CD, 7-13, 11-26
 - CHDIR, 7-13, 11-26
 - CHKDSK, 11-28
 - CLS, 11-32
 - COMMAND, 11-33
 - COMP, 11-35
 - conventions, 11-7
 - COPY, 7-14, 11-36
 - COUNTRY (cconfig.sys), 16-12
 - CSPOOL, 11-42 - 11-43, 13-4
 - DATE, 11-44
 - DEL, 7-15, 11-46
 - DELQ, 11-47
 - DEVICE (cconfig.sys), 16-14
 - DIR, 11-48
 - DISKCOMP, 11-50
 - DISKCOPY, 11-52
 - ECHO (cconfig.sys), 16-16
 - EDITOR, 9-1

- entering, 5-11
- ENVSIZE (cconfig.sys), 16-17
- ERAQ, 11-54
- ERASE, 11-55
- EXE2BIN, 11-56
- EXIT, 11-58
- external, 6-2
- FASTOPEN, 11-59
- FASTOPEN (cconfig.sys), 16-18
- FC, 11-60
- FDISK, 11-62 - 11-67
- FILELINK, 11-68
- FILES (cconfig.sys), 16-19
- FIND, 11-77, 12-4
- FLAGS (cconfig.sys command), 16-20
- FORMAT, 11-79
- getting started, 5-10, 5-12
- GOTO (cconfig.sys), 16-21
- help, 5-12
- IDLE, 11-83, 15-7
- IDLE (cconfig.sys), 16-22
- INSTALL, 1-2
- INSTALL (cconfig.sys), 16-23
- internal, 6-2
- KEYB, 11-84
- KEYS, 11-86
- LABEL, 11-88
- LASTDRIVE (cconfig.sys), 16-24
- learning about, 6-1
- LIMSIZE, 11-89, 15-6
- LOCK, 8-7, 11-90
- LOGOUT, 11-91
- MD, 7-12, 11-94
- MEMSIZE, 11-93, 15-6
- MKDIR, 7-12, 11-94
- MODE, 11-96, 13-16
- MORE, 11-103, 12-4
- NETDRIVE, 11-104, 16-36
- NETWORK (cconfig.sys command), 16-25
- PASSWORD, 8-2, 8-8, 11-105
- PATH, 7-10, 11-108
- PCTERM, 11-109
- PIFED, 11-110
- pipng together, 12-4
- PRINT, 7-16, 11-111, 13-20
- PRINTER, 11-114
- PRINTMAP, 11-115 - 11-116, 13-2, 16-42
- PROMPT, 7-16, 11-117, 16-37
- QUEUES (cconfig.sys command), 16-26
- RD, 7-13, 11-128
- REBOOT, 11-119
- RECOVER, 11-120
- REM (cconfig.sys), 16-25 - 16-27
- REN, 11-122
- RENAME, 7-16, 11-122
- REPLACE, 11-123
- RESTORE, 11-125
- restricting use of, 2-7
- RMDIR, 7-13, 11-128
- searching for at the command line, 11-11
- SECURITY, 11-129
- SET, 11-131, 16-37
- SET (cconfig.sys command), 16-28
- SETUP, 11-133, 16-2
- SHARE, 11-134
- SHOW, 11-135
- SID, 11-136
- SORT, 11-137, 12-4
- specification, 6-5
- STOP, 11-138
- stopping, 6-6
- SUBST, 11-139
- summary of, 11-2
- SUSPEND, 11-140, 15-7, 16-37
- syntax, 11-7
- SYS, 11-141
- TIME, 11-142
- TOUCH, 11-144
- TREE, 11-145
- TYPE, 7-15, 11-146
- types of, 6-2
- VER, 11-147
- VERIFY, 11-148
- VOL, 11-149
- XATTRIB, 8-5, 11-150 - 11-152
- XATTRIB /F, 8-6
- XCOPY, 11-153
- XDEL, 11-156
- XDIR, 11-158
- XSTOP, 11-160 - 11-161
- communications ports
 - see serial ports*

- COMP command, 11-35
- comparing
 - blocks of memory (SID), 14-9
 - diskettes, 11-50
 - files, 11-35
- COMSPEC, 11-131
- concatenating files, COPY command, 11-39
- conditional execution of commands, 10-16, 16-6
- config.cpl, 13-9
 - CLEAR_SCREEN, 13-14
 - DEFAULT_PRIORITY, 13-12
 - FORMFEED, 13-11
 - HOME_CURSOR, 13-14
 - JOB_PRIORITY, 13-12
 - NOFEED_JOBS, 13-12
 - NOT_PRINTER, 13-13
 - TIME_OUT, 13-11
- configuration files
 - autoexec.bat, 16-36
 - cconfig.sys, 16-3
- configuration screens, 1-6
 - search paths, 1-13
 - country/keyboard, 1-6
 - disk cache, 1-12
 - FASTOPEN support, 1-12
 - idle detection, 1-14
 - last drive, 1-13
 - main console, 1-7
 - memory, 1-8
 - moving within, 1-6
 - multigraphics buffers, 1-12
 - NETDRIVE, 1-14
 - open files, 1-13
 - print spooler, 1-12
 - serial ports, 1-14
 - stopping a program, 1-14
 - suspending a program, 1-14
 - switch keys, 1-8
 - system directory, 1-8
 - virtual disks, 1-11
- configuring
 - memory, 1-8
 - operating system, SETUP, 16-1
 - spooler, 13-9, 16-41
- console number, 16-38
- console, copying files to, 11-40
- control keys, 11-11
- conventional memory, D-2
 - definition of, H-4
- converting .exe files, 11-56
- COPY command, 7-14, 11-36 - 11-41
 - see also copying*
- copy-protected applications, 15-20
- copying
 - blocks of text (EDITOR), 9-11
 - directories, 11-153
 - diskettes, 8-17, 11-52
 - files, 7-14, 11-37, 11-123, 11-153
 - files to a console or printer, 11-40
 - master diskettes, 1-20
 - multiple files, 11-153
- country codes, selecting, 1-6, 16-12
- COUNTRY, cconfig.sys command, 16-12
- CPU state, examining, 14-32
- creating
 - batch files, 10-3
 - directories, 7-12, 11-94
 - disk labels, 11-88
 - DOS extended partitions, 11-65
 - DOS partitions, 11-63
 - DOS primary partitions, 11-64
 - files in EDITOR, 9-2
 - files with COPY, 11-41
 - general purpose batch files, 10-4
 - logical drives in an extended partition, 11-65
 - substituted drives, 11-139
- CSPOOL, 16-40
 - command reference, 11-42 - 11-43, 13-4
 - config.cpl, 13-9
 - configuring DOS applications for, 13-14
 - configuring the spooler (*see also* config.cpl), 13-9, 16-41
 - function of, 13-4
 - hints on using, 13-15
 - installing, 1-12, 13-5
 - PRIORITY command option, 13-8
 - QUIT command option, 13-7
 - running, 13-4
 - START command option, 13-7
 - STATUS command option, 13-8
 - STOP command option, 13-9
- currency format, providing, 16-12

- current directory, changing , 11-26
- current disk drive, definition of, H-4
- cursor, definition of, H-4
- cursor, moving in EDITOR, 9-5
- customizing your system, 16-1

D

- data, protecting, 8-1
- databits, definition of, H-4
- date

- changing, 11-44
 - displaying, 11-44
 - format, setting (country), 16-12
 - resetting file date stamp, 11-144
 - setting, 11-44

- DATE command, 11-43

- DEBUG

- see SID*

- debugging tool, 14-1

- decimal numbers (SID), 14-3

- default segment values (SID), 14-35

- default, definition of, H-4

- DEFAULT_PRIORITY, in config.cpl, 13-12

- DEL command, 7-15, 11-46

- deleting

- blocks of text (EDITOR), 9-11
 - characters (EDITOR), 9-8
 - directories, 7-13, 11-128, 11-156
 - disk labels, 11-88
 - DOS partitions, 11-66
 - files with DEL, 7-15, 11-46
 - files with DELQ, 11-47
 - files with ERASE, 11-55
 - lines (EDITOR), 9-9
 - multiple files, 11-156
 - non-DOS partitions, 11-66
 - words (EDITOR), 9-8

- DELQ command, 11-47

- device

- definition of, H-5
 - redirecting output to, 12-3

- device drivers

- additional, 16-29
 - CACHE.EXE, 16-30

- CD ROM, 16-35
- definition of, H-5
- installable, 16-29
- KEYB.COM, 16-31
- MOUSE.SYS, 16-32
- multiport, 16-35
- tape streamer, 16-35
- VDISK.SYS, 16-33

- device names

- redirection, use with, 12-3
 - reserved, 7-3

- DEVICE, cconfig.sys command, 16-14

- DIR command, 5-12, 11-48 - 11-49

- Direct memory access (DMA), 15-16

- directories, 5-4

- access permissions, 2-2, 8-3
 - APPEND command, 11-12
 - backing up, 8-16, 11-20
 - changing, 7-13, 11-26
 - copying, 11-153
 - creating, 7-12, 11-94
 - current, 7-7
 - definition of, 7-6, H-5
 - deleting with RMDIR, 7-13, 11-128
 - deleting with XDEL, 11-156
 - displaying and changing with CHDIR, 7-13, 11-26
 - displaying contents of with DIR, 11-48
 - displaying contents of with XDIR, 11-158
 - displaying tree structure of, 7-12, 11-145
 - listing, 7-12
 - naming, 7-7
 - parent, 7-8
 - password protection, 8-10
 - PATH command, 7-10, 11-108
 - paths, 7-9
 - sharing, 8-3

- DIRECTORY, FILELINK command, 11-75

- dirname, defined, 11-7

- disk cache

- see cache*

- disk operating system

- configuring, 16-1
 - definition of, H-5
 - installing, 1-2
 - transferring (SYS), 11-141

DISKCOMP command, 11-50 - 11-51

DISKCOPY command, 11-52 - 11-53

diskettes

- checking contents of, 8-17
- checking what is on backup, 8-17
- comparing, 11-50
- copying, 8-17, 11-52
- copying masters, 1-20
- definition of, 5-2, H-5
- deleting label on, 11-88
- displaying contents of, 11-48
- displaying label of, 11-149
- displaying tree structure, 11-145
- formatting, 5-6, 11-79
- handling, 5-8, A-45
- modifying label on, 11-88
- preparation of, 5-6
- problems with, A-45
- write-protecting, 5-6

disks

- backing up, 8-13, 11-19
- buffers, 16-11
- cacheing, 11-25, 16-30
- checking condition of, 11-28
- checking what is on backup, 8-17
- creating labels on, 11-88
- definition of, 5-2, H-5
- deleting label on, 11-88
- displaying directory of, 11-48
- displaying tree structure, 11-145
- displaying volume label of, 11-149
- formatting, 11-62
- modifying label on, 11-88
- partitioning, 5-6, 11-62
- preparation of, 5-6
- recovering, 11-120
- sharing, 16-43
- status reports, 11-28
- types supported by FORMAT, 11-79

DISPLAY (SID command), 14-10

DISPLAY COPROCESSOR REGISTERS
(SID command), 14-34

displaying

- 8087 coprocessor registers, 14-34
- contents of memory (SID), 14-10
- date, 11-44

directory contents (XDIR), 11-158

directories and files (DIR), 5-12

directory contents (DIR), 11-48

file contents, 7-15

logical drives in an extended partition, 11-67

messages during startup, 16-16

operating system version number, 11-147

passpoints, 14-18

path of directories (TREE), 11-145

text a screenful at a time, 6-6, 11-103

text file on screen, 11-146

time, 11-142

displays

aligning, 11-98

setting type of, 11-97

DOS

applications, configuring for the spooler, 13-14

applications, running, 15-1

environment, showing (SET), 11-131, 16-28

extended partitions, creating, 11-65

file header, modifying program information,
15-8

partitions, creating, 11-63

partitions, deleting, 11-66

primary partitions, creating, 11-64

DOS session number, 16-38

DOS sessions

amending settings for, 15-6

changing, 5-13

definition of, H-5

number of, 1-7

overview of, 4-4

session numbers as parameters, 16-37

switching between, 1-8

drive letters, assigning, 1-13, 16-24

drivers

see device drivers

drives

changing between, 5-10

definition of, 5-4, H-6

displaying volume label, 11-149

letters, assigning, 1-13, 16-24

specifying, 6-6

substituting, 11-139

types supported by DISKCOPY, 11-52

types supported by FORMAT, 11-79

DUPLICATE, FILELINK command, 11-72
Dynamic Idle Detection, 4-4, 11-83, 15-4, 16-22

E

ECHO

- batch subcommand, 10-12
- cconfig.sys command, 16-16

ECHOERR

- batch subcommand, 10-13

editing

- command line, 11-9
- text files, 9-1

EDITOR

- backup files, 9-12
- command summary, 9-14
- copying blocks of text, 9-11
- cursor control diamond, 9-6
- definition of, H-6
- deleting blocks of text, 9-11
- deleting characters, 9-8
- deleting lines, 9-9
- deleting text, 9-7
- deleting words, 9-9
- entering text in, 9-3
- inserting files at the cursor, 9-7
- inserting text, 9-7
- marking blocks of text, 9-10
- moving blocks of text, 9-10
- moving text down a line, 9-4
- moving the cursor, 9-5
- online help in, 9-4
- overwriting text, 9-7
- quitting, 9-2
- saving files, 9-11
- starting, 9-2
- storing blocks of text in a file, 9-11
- technical details, 9-13

Enter key, definition of, H-6

environment variables

- batch files, 10-6
- SET command, 11-131, 16-28

ENVSIZE, cconfig.sys command, 16-17

ERAQ command, 11-53

ERASE command, 11-55

erasing

- see deleting*

error codes

- BACKUP, 11-21
- RESTORE, 11-126

error messages, A-2

errors, unresolvable, A-46

escape sequences

- ANSI, F-2
- VT52, F-6

EXAMINE CPU STATE (SID command), 14-32

EXE2BIN command, 11-56 - 11-57

executable, definition of, H-6

EXIT command, 11-58

expanded memory, D-8

- definition of, H-6
- page frame, D-8

extended character codes (ECCs), F-5

extended memory, D-6

- definition of, H-6

extended partitions, 11-65

external commands

- definition of, 6-2, H-7
- list of, 6-3

F

FASTOPEN

- cconfig.sys file, 16-18
- command reference, 11-59
- configuring in INSTALL or SETUP, 1-12

FC command, 11-60 - 11-61

FDISK command, 11-62 - 11-67

- Main Menu, 11-62

File Allocation Table (FAT), definition of, H-7

file header, modifying program information, 15-8

file specification, 7-10

FILELINK, 11-68 - 11-76

- baud rates, 11-70
- cable, 11-69

FILELINK commands, 11-68 - 11-76

- DUPLICATE, 11-72

- QUIT, 11-76

- RECEIVE, 11-75
 - SETUP, 11-71
 - SLAVE, 11-72
 - TRANSMIT, 11-73
 - filelists, 7-5
 - filenames, 5-4
 - changing, 7-16
 - definition of, 7-2, H-7
 - names not to be used, 7-3
 - files, 5-4
 - access permissions, 2-2, 8-3
 - attributes, 8-12, 11-16
 - autoexec.bat, 16-36
 - backing up, 8-13, 11-19
 - batch, 10-1
 - cconfig.sys, 16-3
 - combining, 11-39
 - comparing, 11-35
 - copying, 7-14, 11-37, 11-153
 - copying to a console, 11-40
 - copying to a printer, 11-40
 - corrupted, A-46
 - creating with COPY, 11-41
 - creating with EDITOR, 9-2
 - definition of, 7-2, H-7
 - deleting with DEL, 7-15, 11-46
 - deleting with DELQ, 11-47
 - deleting with ERAQ, 11-54
 - deleting with ERASE, 11-55
 - deleting with XDEL, 11-156
 - displaying contents of, 7-15, 11-146
 - filelist support, 11-7
 - filelists, 7-5
 - finding string of characters in, 11-77
 - listing, 7-12, 11-7, 11-48, 11-145, 11-158
 - locking, 11-134
 - moving between directories, 11-122
 - naming, 7-2
 - number open, 1-13, 16-19
 - printing, 7-16, 11-111, 13-20, 16-39
 - recovering, 11-120
 - recovering in EDITOR, 9-12
 - renaming, 7-16, 11-122
 - replacing, 11-123
 - resetting time and date stamps, 11-144
 - restoring backup, 8-16, 11-125
 - SETUP input, 16-3, E-1
 - sharing, 8-3, 11-134, 16-43
 - transferring (FILELINK), 11-68
 - wildcard characters, 7-4
 - FILES, cconfig.sys command, 16-19
 - filespec, definition of, 11-7, H-7
 - FILL (SID command), 14-12
 - filter commands, 11-77, 11-103, 11-137, 12-4
 - FIND command, 11-77 - 11-78, 12-4
 - fixed disks
 - see hard disks*
 - FLAGS, cconfig.sys command, 16-20
 - flags, SID, 14-32
 - floppy disks
 - see diskettes*
 - FOR batch subcommand, 10-14
 - FORMAT command, 11-79 - 11-82
 - drive types supported, 11-79
 - formatting
 - definition of, H-7
 - diskettes, 5-6, 11-79
 - hard disks, 11-62
 - FORMFEED, in config.cpl, 13-11
- ## G
- general purpose batch files, 10-4
 - global
 - passwords, 8-10
 - wildcard, 7-5
 - GO (SID command), 14-13
 - GOTO
 - batch subcommand, 10-15
 - cconfig.sys command, 16-21
 - group class, description of, 8-3
 - group, definition of, H-8
 - groups, 2-2, 8-3
- ## H
- handling
 - diskettes, 5-8, A-45
 - files with wildcards, 7-4
 - hard disks

- backing up, 8-13, 11-19
- definition of, 5-2, H-8
- formatting, 11-62
- maintenance utility, 11-62
- partitioning, 5-6, 11-62
- secure partition, 2-5
- selecting alternate, 11-67

help

- EDITOR, 9-4
- INSTALL, 1-3
- SETUP, 16-2
- SID, 14-7
- switch, 6-5, 11-2

HEXADECIMAL MATH (SID command), 14-14

hexadecimal numbers in SID, 14-3

hidden attribute, 8-12, 11-16

high memory, D-4

- definition of, H-8

HOME_CURSOR, in config.cpl, 13-14

I

I/O, definition of, H-8

IDLE command, 11-83, 15-7

- cconfig.sys file, 16-22

Idle detection, 4-4, 11-83, 15-4, 16-22

IF batch subcommand, 10-16

input

- definition of, H-8
- redirection, 12-2

INPUT, batch subcommand, 10-19

INPUT COMMAND TAIL (SID command), 14-15

INPUTC, batch subcommand, 10-20

inserting

- files at the cursor (EDITOR), 9-7
- remarks in a batch file, 10-23
- remarks in a cconfig.sys file, 16-27
- strings into environment, 11-131, 16-28
- text (EDITOR), 9-7

INSTALL, 1-2

- cconfig.sys command, 16-23
- starting, 1-4

installable device drivers, 16-29

- see also device drivers*

installing

- ARNET.SYS, C-5
- CSPPOOL, 13-5
- FILELINK, 11-72
- operating system, 1-2
- programs at startup, 16-23

internal commands

- definition of, 6-2, H-8
- list of, 6-2

international

- currency format, setting, 16-12
- date and time formats, setting, 16-12

J

JOB_PRIORITY, in config.cpl, 13-12

JOIN command, 11-83

K

KEYB command, 11-83

KEYB.COM, 16-31

keyboard

- defining national, 1-6, 16-31
- locking, 8-7, 11-90
- typematic rate setting, 11-99

KEYS command, 11-86 - 11-87

keys, extended command line editing, 11-10

kilobyte, definition of, H-8

L

LABEL

- command reference, 11-88
- definition of, H-8
- displaying volume, 11-149

last drive, specifying, 1-13, 16-24

LASTDRIVE, cconfig.sys command, 16-24

LIM memory, D-8

- definition of, H-6

LIM page frame

- in expanded memory, D-8

- in TPA, D-9
 - LIMSIZE command, 11-89, 15-6
 - lines
 - setting display number of, 11-99
 - LIST (SID command), 14-16
 - assembly language syntax, 14-38
 - listing
 - directories, 5-12, 7-12, 11-48, 11-145
 - files, 5-12, 7-12, 11-48, 11-145
 - LOAD PROGRAM (SID command), 14-11
 - loading
 - second command.com, 11-33
 - programs with INSTALL, 16-23
 - LOCK command, 8-7, 11-90
 - locking keyboard and screen, 8-7, 11-90
 - log file, BACKUP command, 11-20
 - logging in, 5-9
 - as Superuser, 1-19
 - logical drives
 - creating in an extended partition, 11-65
 - definition of, H-9
 - displaying in extended partition, 11-67
 - LOGOUT command, 11-91
 - lost clusters, 11-28
- ## M
- making
 - batch files, 10-3
 - directories, 7-12, 11-94
 - general purpose batch files, 10-4
 - marking text blocks (EDITOR), 9-10
 - master diskettes, copying, 1-20
 - MD command, 7-12, 11-94 - 11-95
 - megabyte, definition of, H-9
 - memory
 - allocation to applications, 15-3
 - banked, D-7
 - comparing blocks of (SID), 14-9
 - configuring, 1-8
 - conventional, D-2
 - definition of, H-9
 - description of, 5-5, D-2
 - expanded, 5-5, D-8
 - extended, D-6
 - filling an area with byte or word constant, 14-12
 - high, D-4
 - LIM, D-8
 - LIM allocation, 15-3
 - listing contents of in assembly language, 14-16
 - management, 4-5, D-1
 - moving blocks of data values in, 14-17
 - non-banked, D-10
 - paged, D-7
 - sharing, 16-42
 - standard, 5-5
 - transient program area, D-3
 - types of, 5-5, D-2
 - upper, D-5
 - memory disks
 - see virtual disks*
 - MEMSIZE command, 11-92, 15-6
 - menu, sample batch file for, G-1, G-3
 - MKDIR command, 7-12, 11-94 - 11-95
 - mode, screen (ANSI.SYS), F-4
 - MODE command, 11-96 - 11-102, 13-16
 - aligning the display on a CGA monitor, 11-98
 - assigning printers, 11-100, 13-19
 - setting keyboard typematic rate, 11-99
 - setting lines and columns, 11-99
 - setting display type, 11-96
 - setting up communications ports, 11-101, 13-17
 - setting up printers, 11-101, 13-16
 - setting up serial ports, 13-17
 - modem, definition of, H-9
 - MORE command, 11-103, 12-4
 - mouse, 1-7, 15-14, 16-32
 - MOUSE.SYS, 16-32
 - MOVE (SID command), 14-17
 - moving
 - blocks of text (EDITOR), 9-10
 - files between directories, 11-122
 - text down a line (EDITOR), 9-4
 - multigraphics, configuring for, 1-12
 - multiport boards, 1-14, C-1
 - switch settings, C-1
 - multitasking
 - definition of, H-9
 - description of, 4-3
 - multiuser environment
 - cconfig.sys, 15-17

- copy-protected applications, 15-20
- definition of, H-10
- description of, 4-2
- direct memory access (DMA), 15-16
- mouse, 15-14
- multiple copies, running, 15-18
- networking, 15-7
- printing, 15-4
- terminals, setting up, 15-13
- TSRs, 15-17

N

- NETDRIVE command, 11-104
 - autoexec.bat file, 16-36
- NETWORK, cconfig.sys command, 16-25
- networking, 11-104, 15-7
 - overview of, 4-3
- NOFEED_JOBS, in config.cpl, 13-12
- non-banked memory, D-10
- NOT_PRINTER, in config.cpl, 13-13
- null modem cable, B-2

O

- online, definition of, H-10
- open files, specifying number of, 1-13, 16-19
- operating system
 - utilities, overview of, 4-5
 - basic concepts, 5-2
 - configuration files, 16-3
 - customizing, 16-1
 - default command processor, 11-33
 - displaying version number, 11-147
 - installable device drivers, 16-29
 - installing, 1-2
 - overview of, 4-1
 - security, 2-1, 8-1
 - system files, transferring, 11-141
- OS environment variable, 10-6
- output
 - appending, 12-2
 - definition of, H-10
 - redirecting, 12-2

- overwriting text (EDITOR), 9-7
- owner class, description of, 8-3

P

- paged memory, D-7
- parallel ports
 - changing assignments, 11-115
 - displaying current settings for, 13-2
 - mapping to logical device names, 13-2, 16-41
 - sharing, 16-41
- parallel, definition of, H-10
- parent directory, 7-8
- parity, definition of, H-10
- partitions, 5-6, 11-62
 - creating DOS extended, 11-65
 - creating DOS primary, 11-64
 - creating logical drives in extended, 11-65
 - creating on hard disks, 11-63
 - definition of, H-10
 - deleting DOS, 11-66
 - making secure, 2-5
 - maximum size of, 11-62
 - selecting bootable, 11-66
- PASSPOINT (SID command), 14-18
- passpoints (SID), 14-18
- PASSWORD command, 8-2, 8-8, 11-105 - 11-107
- passwords, 2-2, 8-1, 11-105
 - assigning, 8-9, 11-105
 - changing, 8-2, 11-105
 - definition of, H-11
 - global, 8-10, 11-106
 - issuing, 8-9
 - levels of protection, 8-8
 - protecting directories with, 8-10, 11-105
 - protecting files with, 11-105
 - removing, 8-11, 11-106
- PATH command, 7-10, 11-108
- paths
 - APPEND command, 11-12
 - definition of, H-11
 - understanding, 7-9
- PAUSE batch subcommand, 10-21
- PAUSEERR batch subcommand, 10-22

pausing text on screen, 6-6
PCTERM, 3-1, 11-109
performance of application, improving, 15-1
peripheral, definition of, H-11
PIFED command, 11-110
pin connections, FILELINK cable, 11-70
piping, 12-4
 definition of, H-11
ports
 see parallel ports
 see serial ports
primary partition, creating, 11-64
PRINT command, 11-111 - 11-113, 13-20
print queue, 11-112, 13-21
print spooler
 see CSPOOL
PRINTER command, 11-114
printers
 assigning, 11-100, 13-19
 changing assignment, 11-115, 13-2
 copying files to, 11-40
 displaying current port settings, 13-2
 listing, 11-116
 mapping ports to logical device names, 11-115, 13-2, 16-41
 setting up, 11-101, 13-16
printing, 13-1, 16-39
 configuring DOS applications, 13-14
 files, 7-16, 11-111, 13-20
 in multiuser environment, 15-4
 print queue, 11-112, 13-21
PRINTMAP command, 11-115 - 11-116, 13-2, 16-42
PRIORITY, CSPOOL command option, 13-8
problems, A-44
 error messages, A-2
 with diskettes, A-45
program information, modifying with PIFED, 15-8
prompt, 5-10
 changing, 7-17, 11-117
PROMPT command, 7-16, 11-117 - 11-118
 autoexec.bat file, 16-37
protecting
 data, 8-1

directories with passwords, 8-10, 11-105
files with passwords, 8-8, 11-105

Q

QUERY I/O (SID command), 14-21
queue, print, 11-112, 13-21
QUEUES, cconfig.sys command, 16-26
QUIT, CSPOOL command option, 13-7
quitting
 EDITOR, 9-2
 FILELINK, 11-76
 SID, 14-2

R

RAM disks and RAM drives
 see virtual disks
RAM, definition of, 5-5, D-2, H-11
RD command, 7-13, 11-128
READ (SID command), 14-23
read-only attribute, 8-12, 11-16
read-only, definition of, H-12
reading in a file (EDITOR), 9-7
real time, definition of, H-12
REBOOT command, 11-119
RECEIVE, FILELINK command, 11-75
receiving files from a slave computer (FILELINK), 11-75
RECOVER command, 11-120
redirecting
 input, 12-2
 output, 12-2
 output to or from a device, 12-3
 and pipes, 12-6
refreshing screen display, 11-32
REM
 batch subcommand, 10-23
 cconfig.sys command, 16-25, 16-27
removing
 directories, 7-13, 11-128, 11-156
 files with DEL, 11-46
 files with DELQ, 11-47
 files with ERASE, 11-55

- passwords, 8-11, 11-105
- REN command, 7-16, 11-122
- RENAME command, 7-16, 11-122
- renaming
 - files with RENAME, 7-16
 - files with XCOPY, 11-154
- repetitive execution of commands (FOR), 10-14
- REPLACE command, 11-123 - 11-124
- replaceable variables (SHIFT), 10-24
- reports, disk status, 11-28
- reserved device names, 7-3
- resetting, file time and date stamps, 11-144
- resident commands
 - see internal commands*
- resource sharing, 4-2, 16-39
- RESTORE command, 8-16, 11-125 - 11-127
 - error codes, 11-126
- restoring backup files, 8-16, 11-125
- RMDIR command, 7-13, 11-128
- ROM, definition of, H-12
- root directory, 7-6
 - definition of, H-12

S

- screen display
 - clearing, 11-32
 - refreshing, 11-32
- screen modes, ANSI.SYS, F-5
- screen, locking, 8-7, 11-90
- scrolling
 - definition of, H-12
 - stopping, 6-6
- search
 - for a string of characters (FIND), 11-77
 - path (APPEND), 11-12
 - path (PATH), 11-108
- SEARCH (SID command), 14-26
- search path, setting, 11-108
- searching for commands at the command line, 11-11
- security, 2-1, 8-1
 - locking keyboard and screen, 11-90
 - see also system security*
- SECURITY command, 11-129 - 11-130

- selecting
 - alternate hard disks, 11-67
 - bootable partitions, 11-66
- sending files with TRANSMIT (FILELINK), 11-73
- serial cable
 - connecting a serial terminal, B-2
 - for FILELINK, 11-69
- serial ports
 - changing assignment, 11-116
 - configuring, 1-14
 - displaying current settings for, 13-2
 - mapping to logical device names, 13-2, 16-41
 - setting up (MODE), 11-101, 13-17
 - sharing, 16-41
- serial terminal, connecting, B-1
 - null modem cable, B-2
- serial, definition of, H-12
- session, definition of, H-12
- SET
 - cconfig.sys command, 16-28
 - (SID command), 14-24
 - command, 11-131 - 11-132
 - autoexec.bat file, 16-37
- SETMEM statements, E-4
- SETMISC statements, E-5
- SETMULTI statements, E-6
- SETPORT statements, E-7
- SETTERM statements, E-8
- setting
 - keyboard typematic rate (MODE), 11-99
 - lines, columns (MODE), 11-98 - 11-99
 - screen display (MODE), 11-96
- SETUP command, 1-6, 11-133, 16-2
 - help, 16-2
 - input file, 16-3
- setup input file, E-1
 - SETMEM statements, E-4
 - SETMISC statements, E-5
 - SETMULTI statements, E-6
 - SETPORT statements, E-7
 - SETTERM statements, E-8
 - statement format, E-2
- SETUP, FILELINK command, 11-71
- SHARE command, 11-134
- sharing resources, 4-2, 16-39

- SHIFT batch subcommand, 10-24
- SHOW command, 11-135
- SID, 14-1
 - addresses, specifying 20 bit, 14-3
 - ASCII characters, use of, 14-3
 - assembly language syntax for A and L
 - commands, 14-38
 - character values, 14-3
 - command summary, 14-5
 - command tail, parsing of, 14-15
 - decimal numbers, 14-3
 - default segment values, 14-35
 - definition of, H-12
 - error messages, A-44
 - hexadecimal numbers, 14-3
 - operators in expressions, 14-4
 - quitting, 14-2
 - starting, 14-2
- SID command, 11-136
- SID commands
 - ?, 14-7
 - ??, 14-7
 - A (ASSEMBLE), 14-8
 - B (BLOCK COMPARE), 14-9
 - D (DISPLAY), 14-10
 - E (LOAD PROGRAM), 14-11
 - F (FILL), 14-12
 - G (GO), 14-13
 - H (HEXADECIMAL MATH), 14-14
 - I (INPUT COMMAND TAIL), 14-15
 - L (LIST), 14-16
 - M (MOVE), 14-17
 - P (PASSPOINTS), 14-18
 - Q (QUERY I/O), 14-21
 - Q (QUIT), 14-21
 - R (READ), 14-23
 - S (SET), 14-24
 - SR (SEARCH), 14-26
 - summary, 14-5
 - T (TRACE), 14-27
 - U (UNTRACE), 14-29
 - V (VALUE), 14-30
 - W (WRITE), 14-31
 - X (EXAMINE CPU STATE), 14-32
 - Z (display 8087 coprocessor registers), 14-34
- SLAVE, FILELINK command, 11-72
- SORT command, 11-137, 12-4
- stamps, time and date, 11-44, 11-144
- START, CSPOOL command option, 13-7
- starting
 - applications from autoexec.bat, 16-38
 - EDITOR, 9-2
 - FDISK, 11-62
 - INSTALL, 1-4
 - second copy of COMMAND.COM, 11-33
 - SID, 14-2
- station number, 16-38
- status line, 1-7
- STATUS, CSPOOL command option, 13-8
- STOP command, 11-138
- STOP, CSPOOL command option, 13-9
- stopbits
 - definition of, H-13
- stopbits, definition of, H-13
- stopping
 - batch files, 10-4
 - commands, scrolling, 6-6
 - EDITOR, 9-2
 - FILELINK, 11-76
 - programs running (BREAK), 11-24, 16-10
 - programs running (REBOOT), 11-119
 - SID, 14-2
- storing text blocks in a file (EDITOR), 9-11
- string, definition of, H-13
- strings, searching for (FIND), 11-77
- subdirectories, 7-6
- subdirectories, displaying directory of, 11-48
- subdirectory, definition of, H-13
- SUBST command, 11-139
- Supergroup, 1-18
 - definition of, H-13
- Superuser, 1-18, 2-1
 - adding users to system, 2-3
 - commands, restricting use of, 2-7
 - definition of, H-13
- SUSPEND command, 11-140, 15-7
 - autoexec.bat file, 16-37
- switch, definition of, 6-5
- switch keys, 1-8
- syntax of commands, 11-7
- SYS command, 11-141
- system attribute, 8-12, 11-16

- system configuration, changing defaults of, 16-1
- system disk, creating (SYS), 11-141
- system prompt, 5-10
 - changing, 7-17, 11-117
 - definition of, H-13
- system reset (REBOOT), 11-119
- system security, 2-1, 8-1, 11-129 - 11-130
 - access permissions, 2-2, 8-3
 - adding users, 2-3
 - passwords, 8-2
 - user names, 2-2

T

TAB key in EDITOR, 9-4

tables

- ANSISYS display modes, F-4
- ANSISYS screen modes, F-5
- BACKUP error codes, 11-21
- Batch subcommands, 6-4
- Batch subcommands summary, 10-7
- cconfig.sys commands, 16-5
- Command line editing keys, 11-10
- Command set, 11-2
- Common filename extensions, 7-3
- COPY device names, 11-40
- Country codes, 16-13, 16-31
- CSPOOL command options, 13-6
- CSPOOL commands, 11-42
- CSPOOL entries in config.cpl, 13-9
- DISKCOPY, drives supported by, 11-53
- EDITOR commands, 9-14
- External commands, 6-3
- FILELINK, baud rates, 11-71
- FORMAT, drives supported by, 11-79
- Installable device drivers, 16-29
- Internal commands, 6-2
- National keyboards, 11-84
- PCTERM configuration settings, 3-4
- Redirection, device names with, 12-3
- Reserved device names, 7-3
- RESTORE error codes, 11-126
- SETMEM settings, E-4
- SETMISC settings, E-5
- SETMULTI settings, E-6

- SETPORT settings, E-7
- SETTERM settings, E-8
- SHOW options, 11-135
- SID commands summary, 14-5
- SID, default segment values, 14-37
- SID, flag name abbreviations, 14-32
- STOP display information, 11-160
- VT52 escape sequences, F-6

terminal emulator, 3-1, 11-109

terminals

- adding, 3-1, 4-2
- serial, connecting, B-1
- setting up, 15-13
- station numbers as parameters, 16-37

Terminate-and-Stay-Resident (TSR)

applications, 15-17

text blocks, in EDITOR, 9-10

text files, editing, 9-1

time

- changing, 11-142
- format, setting (country, 16-12
- resetting file time stamp, 11-144
- setting, 11-142

time and date stamping, 11-44

TIME command, 11-141

TIME_OUT, in config.cpl, 13-11

timeout, definition of, H-13

TOUCH command, 11-144

TPA, D-3

TRACE (SID command), 14-27

transferring

- DOS system files, 11-141
- files (FILELINK), 11-68

Transient Program Area (TPA), D-3

TRANSMIT, FILELINK command, 11-73

tree

- directory structure, 7-6
- duplicating on another disk (XCOPY), 11-154

TREE command, 7-12, 11-145

troubleshooting checklist, A-44

TSR applications, 15-17

TYPE command, 7-15, 11-146

type-1 segment value (SID), 14-35

type-2 segment value (SID), 14-36

typematic rate, setting, 11-99

U

unresolvable errors, A-46
UNTRACE (SID command), 14-29
updating previous versions of files, 11-123
upper memory, D-5
 definition of, H-14
user names, 2-2
users
 adding to system, 2-3, 11-129
 classes of, 8-3
 deleting from system, 11-129
 editing details of, 11-129
 listing details of, 11-129
utilities, overview of, 4-5
utility, definition of, H-14

V

VALUE (SID command), 14-30
variables
 (SHIFT), 10-24
 batch files, use in, 10-4
 definition of, H-14
 environment, 10-6
VDISK.SYS, 16-33
VER command, 11-147
VER environment variable, 10-6
VERIFY command, 11-148
version of operating system, displaying, 11-147
video memory, definition of, H-14
virtual 8086 mode, D-7
virtual consoles
 see DOS sessions
virtual disks
 configuring in INSTALL or SETUP, 1-11
 description of, 5-4
 definition of, H-14
 setting up (VDISK.SYS), 16-33
VOL command, 11-149
volume labels
 characters allowed in, 11-88
 creating, 11-88
 definition of, H-14
 deleting, 11-88

 displaying, 11-149
 modifying, 11-88
VT52 escape sequences, F-6

W

wildcard characters
 backing up, use in, 8-16
 definition of, H-14
 in files, 7-4
wildspec, definition of, 11-7, H-14
world class, description of, 8-4
WRITE (SID command), 14-31
write-protect notch, 5-7
write-protecting
 definition of, 5-6, H-15
 diskettes, 5-6

X

X (SID command), 14-32
XATTRIB command, 8-5, 11-150 - 11-152
XDEL command, 11-156 - 11-157
XDIR command, 11-158 - 11-159
XSTOP command, 11-160 - 11-161

Z

Z (SID command), 14-34

NOVELL

2015-2004-002

70 GARDEN COURT

MONTEREY, CA
93940

(408) 649-3896
(800) 649-0750

MADE IN
U.S.A.



NOVELL