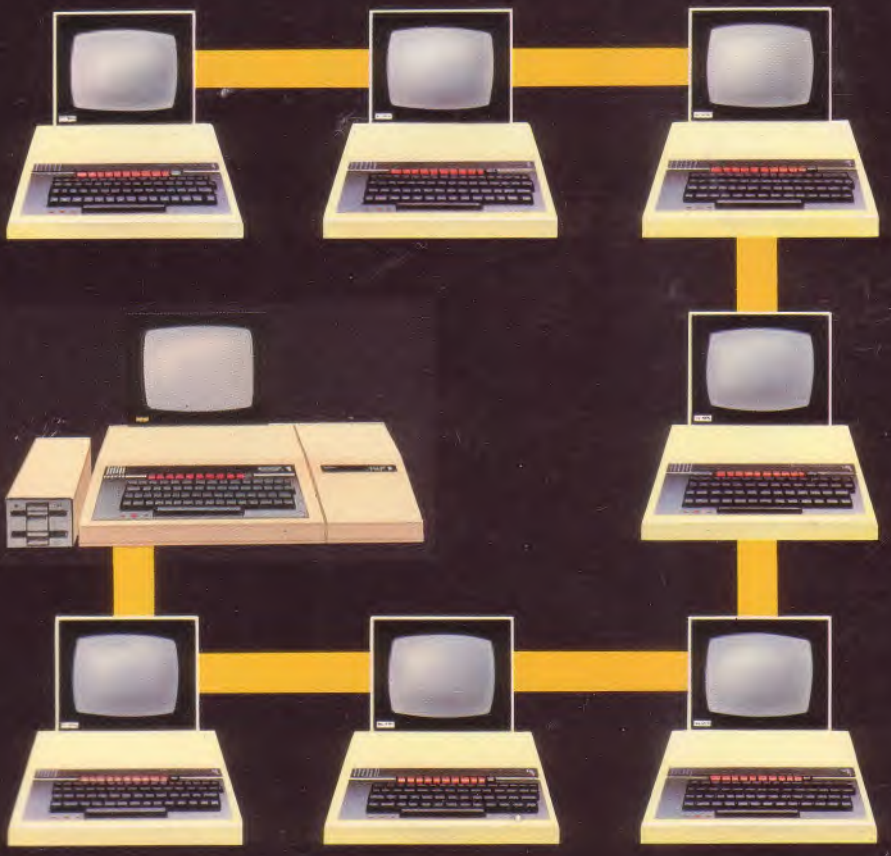
ACORN COMPUTER

Econet
level 2 file server
USER GUIDE



Acorn
Econet

Level 2 File Server user guide

Within this publication the term "BBC" is used as an abbreviation for "British Broadcasting Corporation"

412,018 Issue 1
October 1983

Written and designed by Baddeley Associates, Cambridge

The product described in this manual and products for use with it, are subject to continuous developments and improvement. All information of a technical nature and particulars of the product and its use (including the information in this manual) are given by Acorn Computers in good faith. However, it is acknowledged that there may be errors or omissions in this manual. A list of details of any amendments or revisions to this manual can be obtained upon request from Acorn Computers Technical Enquiries. Acorn Computers welcome comments and suggestions relating to the product and this manual.

All correspondence should be addressed to:

Technical Enquiries
Acorn Computers Limited
Fulbourn Road
Cherry Hinton
Cambridge
CBI 4JN

All maintenance and service on the product must be carried out by Acorn Computers' authorised dealers. Acorn Computers can accept no liability whatsoever for any loss or damage caused by service or maintenance by unauthorised personnel. This manual is intended only to assist the reader in the use of the product, and therefore Acorn Computers shall not be liable for any loss or damage whatsoever arising from the use of any information or particulars in, or any error or omission in, this manual, or any incorrect use of the product.

Contents

Using Econet

Econet is a network — a set of computers connected together. Each computer on the network is called a station. This guide tells you how you can use Econet from your own station to communicate with other computers on the network.

The most important station you'll be communicating with is the file server. This is a computer attached to a device that can record programs and data on to floppy discs. Floppy discs are like gramophone records, except that:

- they are smaller
- they are flexible, not rigid
- you can record on to them as well as play back from them.

Through Econet, you can send work — for example, a program you have written — to the file server, and ask it to store the program on one of its discs. You can also instruct the file server to retrieve what you've stored and put it back into your computer's memory: so, if you wanted to, you could get that program back and run it again. Storing work is called saving; bringing it back is called loading. With Econet, all the users on the network can share the same file server, so they can save or load work whenever they want to.

Your network may also include a computer connected up to a printer: this is a printer server. If your Econet includes a printer server, you will also be able to use the network to print out copies of your work.

There will be a person looking after the day-to-day running of your network — the network manager. If you have difficulties, the network manager is the person to see.

This guide begins by explaining how you start a session of work on the network, and then takes you through the rules for saving and loading. After that, we explain all the other instructions you can send to the file server and the printer server.

Using commands

Throughout this guide there are instructions for typing in commands to which the following rules apply:

- showing a word in [ ] means it is a key

  EXAMPLE: [RETURN] means the RETURN key

- descriptions in < > should be replaced by the information required, without typing the brackets

  EXAMPLE: <filename> means type the name of a file

- type characters not in brackets exactly as they are shown

When you type in a command the computer will not carry it out until you press [RETURN].

Most of the Econet commands can be abbreviated, to save you time when you get used to using them. The abbreviations are shown in the reference section at the back of this guide.

Getting going

Switching on

Switch on your BBC micro and your monitor.

Screen: **BBC Computer 32K**
Econet Station xxx
BASIC

xxx is the number of your station on the network

You are now ready to use Econet.

If the message "Econet Station" is missing, hold down N while pressing then releasing [BREAK]; or hold down [CTRL] and N together while pressing and releasing [BREAK].

If the "Econet Station" message is still missing, tell your network manager. If there is a "No clock" message, tell the network manager straight away.

Logging on *I AM

Whenever you start an Econet session, you have to identify yourself to the file server, by typing in your identifier. If you are in doubt about what yours should be, ask your network manager.

EXAMPLES
RPJ
JULIE
FORM3

To start working with the file server type:

***I** AM **<identifier>[RETURN]**

If you get a message "User not known", ask your network manager for help.

When logging on, you may have to specify the file server you want to log on to. You will have to do this if your network has more than one file server, or if your file server has a different station number from the one Econet expects.

Type: ***I** AM **<number><identifier>[RETURN]**

Your network manager will tell you exactly what number to type.

Passwords * PASS

You can choose a password for yourself, which you type in when logging on. This makes it impossible for other users who don't know your password to log on using your identifier, and so gain access to your files. Your password can be up to six characters long, and can include letters and numbers. It is important not to tell any other user what your password is — and to avoid passwords that would be easy for other users to guess.

To set your password, log on, and

type: *PASS "" **<your password>[RETURN]**

From now on, you will have to give your password whenever you log on, by
typing ***I** AM **<identifier><password>[RETURN]**

EXAMPLE

Type: *I AM ROBERT ACORN[RETURN]

You may be able to prevent your password appearing on the screen (where people looking over your shoulder might see it) by going through this procedure:

Type: ***I** AM **<identifier> :[RETURN]**
<password>[RETURN]

Your password will not appear as you type it in.

NOTE: this facility may not be available on your Econet. See the notes on software versions in the reference section.

If you type in a different password, or no password at all, you will get a "Wrong password" error, and the file server will not allow you to log on.

You can change your password at any time, using the *PASS command.

Type: ***PASS <current> <new> [RETURN]**

<current> here stands for your current password;
<new> for your new password.

If you forget your password, ask the network manager for help.

Finishing

You can now start work. When you want to end your session, you log off.

Type: ***BYE[RETURN]**

Simple filing

Saving and loading BASIC programs — SAVE and LOAD

If you want to store a BASIC program on the file server so that you can use it again later

type: **SAVE"<filename>"[RETURN]**

A copy of the BASIC program in your computer's memory will be taken and stored for you on the file server. This copy is called a file, and is stored with the filename you typed in.

EXAMPLE

Type: SAVE"TUESDAY"[RETURN]

To get a BASIC program back from disc into your computer

type: **LOAD"<filename>"[RETURN]**

EXAMPLE

Type: LOAD"MATHS"[RETURN]

To run the program, type RUN as usual.

To load and run a BASIC program in one operation

type: **CHAIN"<filename>"[RETURN]**

EXAMPLE

Type: CHAIN"BASIC"[RETURN]

If, when using SAVE, you specify a filename that is the same as an existing filename, the new program will replace the old one. This is useful if you are writing a new version of an old program, but frustrating if you delete the old file unintentionally. We explain later how to protect your files against accidental deletion.

NOTE: your computer holds one program at a time as its current program. If you press the [BREAK] key, that program is cleared from its memory. The program can be recovered by

typing: **OLD[RETURN]**

If you SAVE the program before recovering it, an empty file will be created. It's a good idea to check that the program is there — by typing LIST, for example — before saving it.

Displaying a catalogue *CAT

To get a list of your files on the screen

type: ***CAT[RETURN]**

Screen:
JULIE (039) Public
Master-disc Option 00 (Off)
Dir. JULIE Lib. LIBRARY

BASIC WR/ MATHS WR/
TUESDAY WR/

The information in your catalogue is split into:

- a header
- the list of files.

The files in this catalogue are BASIC, MATHS and TUESDAY. The letters after these names show your rights of access to the files — in this example, you can both write to (W) and read (R) each file. Access rights are explained in the section of this guide on Protecting your files.

The header shows:

1 The name of the directory whose contents are listed below

2 The directory's current cycle number (this is a number that changes each time you change anything in the directory)

3 The name of the disc you are logged on to

4 The name of your currently selected directory

5 Your rights of access to the files listed (owner or public)

6 The autostart option you have selected

7 The name of the directory currently selected as your library

The meanings of these pieces of information will become clear as you work through this guide and get used to using the file server.

Naming files

There are rules for the names you give your files. Filenames can have up to ten characters and can use any combination of letters and numbers. You can also use:

! % & = - | \ { _ + ; } 7 < > ? /

Do not use any other symbol; spaces are also not allowed.

EXAMPLES

MATHS	MYPROG
TUESDAY	BUG!!!
BASIC	MONDAY1NOV

You can mix upper and lower case letters so that MATHS, Maths and maths will all refer to the same file.

If you try to give a file a filename that breaks these rules, the filing system will respond with the message " Bad file name".

Files and directories

Files in the Econet filing system are organised into directories. When you type *CAT, the list of names that comes up on your screen is your directory.

Directories can include other directories as well as files. When a catalogue listing includes a directory, the letter "D" appears next to it.

EXAMPLE

TUNES DL/

To create your own directories, you will need the command *CDIR, which is explained in the section of this guide on *Using* directories.

Deleting files *DELETE

Type: ***DELETE <filename>[RETURN]**

and your file of that name will be permanently removed from the disc. If, however, the file is protected, an "Entry locked" message will appear on your screen, and you will not be able to delete it.

The section below on Protecting your files explains how to use the *ACCESS command to lock a file in this way.

To delete a sub-directory:

- unlock all the files it contains, using *ACCESS
- delete all those files, using *DELETE
- delete the directory, using *DELETE.

The *Using* directories section of this guide explains how to specify files inside sub-directories so that you can unlock and delete them. You will not be able to delete a directory if someone else on your network is using it.

Renaming files * RENAME

You can change the name of any of your files.

Type: * **RENAME <current> <new> [RETURN]**

<current> here stands for the current filename;
<new> for the new filename.

EXAMPLE

*RENAME TRYOUT WORK

This changes the *name of* the *file* TRYOUT to WORK.

You cannot rename directories, but you can move a file from one directory to another, provided they are both on the same disc. How to do this is explained in the Using directories section of this guide.

You may get a message "Entry locked". This means the file is has been protected using *ACCESS, as explained in the next section.

Protecting your files *ACCESS

You may want to protect your files so that other users cannot interfere with them, and so that you cannot accidentally erase them. This process is called setting an access string. The command you use is *ACCESS.

You protect your file by typing ***ACCESS** followed by

- the filename
- the rights of access to the file that you want to have yourself — the owner access rights
- an oblique stroke character /
- the rights of access you want to allow other users to have to your files — the public access rights
- [RETURN]

You specify the kind of access by using the letters W, R and L.

W means **write** — the file can be written to. The only way to write to files is by random access writing — a method described in the section of this guide on Reading and writing to files. Unless you specify W, it will not be possible to write to the file.

R means **read** — the file can be read.

L means **locked** — the file cannot be deleted. L is used only to protect your files against accidental deletion by you: your files are automatically locked to other users.

The important access letters at this stage are R and L. Their effects are summarised in this table:

without R you cannot: with L you cannot:

LOAD the file	SAVE over the file
*LOAD it	*DELETE it
*<filename> it	*RENAME it
*EXEC it	create a *SPOOL file over it
	OPENOUT it

(*LOAD, *<filename>, *EXEC and *SPOOL are explained in the section on Machine-code and data files; OPENOUT is explained in the section on Reading and *writing* to files.)

If you try to read a file which doesn't have R in its access string, an "Insufficient access" message will come up on the screen. If you try to delete a file which has an L in its access string, you will get an "Entry locked" message.

EXAMPLES

*ACCESS MATHS LWR/R

means you will be able to write to and read your file MATHS, but not to delete it; and other users *will* be able to read it, but not write to it. You will get a "Bad attribute" error message if you specify L in the right hand half of the access string: files are automatically locked to other users.

*ACCESS MATHS R/

means you *will* be able to read and delete MATHS but not write to it. Other users *will* have no access to it at all.

You can use *ACCESS to unlock and re-lock your directories, but W and R have no meaning in access strings for directories.

Until you specify access strings for your files, the file server will automatically give them the access string WR/, which means you will be able to delete, write to and read them, and that others will have no access to them at all.

Getting information about your files

* INFO, *EX and *OPT 1

The *INFO command displays information about the file so that you can find out how big it is, what its access string is, and what its reload and execution addresses are (these terms are explained in the section on Machine-code and data files).
Type: ***INFO <filename>[RETURN]**

The information is shown in this order:

- filename
- reload address
- execution address
- size
- access string
- the date the file was most recently saved
- the System Internal Name of the file: this gives the location of the file on the disc.

The reload and execution addresses, the file size, and the System Internal Name are given in hexadecimal.

EXAMPLE
Typing: *INFO BASIC[RETURN]

might display

BASIC FFFF1200 FFFF8023 000043 WR/
01:12:85 000145

Here, the display shows:

- filename: BASIC
- reload address: FFFF1200
- execution address: FFFF8023
- size: 000043
- access string: WR/
- the date the file was most recently saved: 1 December 1985
- the System Internal Name of the file: 000145.

You can use *INFO on directories as well as on files. The reload and execution addresses, which have no meaning for directories, will show as zeros.

The *EX command makes the computer perform an *INFO on all the items in your directory. The result is a display that looks like a *CAT display, but with the extra *INFO details on each file.
Type: ***EX[RETURN]**

EXAMPLE

Type: *EX[RETURN]

Screen: ROBERT (014) Owner
Master-disc Option 00 (Off)
Dir. ROBERT Lib. LIBRARY

BASIC FFFF1200 FFFF8023 000043
WR/ 01:12:85 000145
TUNES 00000000 00000000 000200 DL/
03:12:85 000153

The directory ROBERT contains, in this example, a file called BASIC and a directory called TUNES.

The *OPT1 command can be used to get on-screen information about your files as you save or load them.

***OPT1,1** turns this facility on. After entering this command, file details will come up every time you save or load.
***OPT1,0** turns the facility off.

EXAMPLE

Type: *OPT1,1[RETURN] LOAD"
MATHS"[RETURN]

Screen: MATHS FFFF1200 FFFF8023 000038

Filing system commands in BASIC programs

You can include Econet file server commands in your BASIC programs.

EXAMPLE

This is a very simple program to get a repeating catalogue on your *screen.*

```
10 REM CATALOGUE 20
*CAT
30 GOTO 20
40 END
```

Using directories

Creating directories *CDIR

You can create sub-directories within your main directory, so that you can organise your files into groups and sub-groups. For example, you might put all your programs that play tunes into a sub-directory called TUNES. The ones that play film tunes could go into a sub-directory of TUNES called FILM.

Before you can organise your files in this way, you have to create the directories you want, using the *CDIR command.

To create a directory

type: ***CDIR <directory name>[RETURN]**

EXAMPLES

*CDIR PROGS
*CDIR TUNES

If you now type *in* *CAT, you will see PROGS and TUNES listed among the *entries in* your main directory. Next to them *will* be the letters DL/, to show that

- they are directories
- they are locked.

Pathnames for files

To use files that are in sub-directories, you will need to specify the sub-directory name as well as the filename. You do this by giving the sub-directory name first, then a full stop, then the filename.

EXAMPLES

TUNES.A MATHS.
MONDAY

Names like this are called pathnames.

You may have directories within your sub-directories: you can have as many layers of directories as you like, up to 255. To specify a pathname for a file:

- start with the sub-directory name
- then list all the further directories, in order
- end with the filename
- separate all the elements with a full stop.

EXAMPLES

TUNES.FILM.STING PHIL.
PROGS.BASIC.SQUARES

Every command that takes a filename can take a file pathname.

EXAMPLE

If STING is a file in FILM, which is a directory within the sub-directory TUNES, typing LOAD"STING" would produce the error message "Not found".

To load the file STING

type: LOAD"TUNES.FILM.STING"[RETURN]

Pathnames for directories

Your directories have pathnames too.

To specify a pathname for a directory:

- start with the sub-directory name
- then list all the further directories, in order
- end with the directory name
- separate all the elements with a full stop.

EXAMPLES

TUNES.FILM PHIL.
PROGS.BASIC

Every command that takes a directory name can take a directory pathname.

EXAMPLE

To set up two directories POP and FILM within the sub-directory TUNES

type: *CDIR TUNES.POP[RETURN]
*CDIR TUNES.FILM[RETURN]

Displaying a sub-directory *CAT

*CAT produces a display of your main directory; you can get displays of your sub-directories by typing ***CAT <directory name>[RETURN]**

EXAMPLE

You log on as ROBERT.

Type: *CAT[RETURN]

Your main directory comes up on the *screen*. It contains two ordinary files and a sub-directory called TUNES.

```
Screen: ROBERT (013)          Owner
        Master-disc           Option 00 (Off)
        Dir. ROBERT           Lib. LIBRARY
        BASIC WR/WR           MATHS WR/
        TUNES DL/
```

Type: *CAT TUNES[RETURN]

```
Screen: TUNES (002)           Owner
        Master-disc           Option 00 (Off)
        Dir. ROBERT           Lib. LIBRARY
        A       WR/           B       WR/
        C       WR/           FILM    DL/
        POP     DL/
```

The directory TUNES contains three ordinary *files* (each is a program that plays a tune) *and* two further directories, FILM and POP.

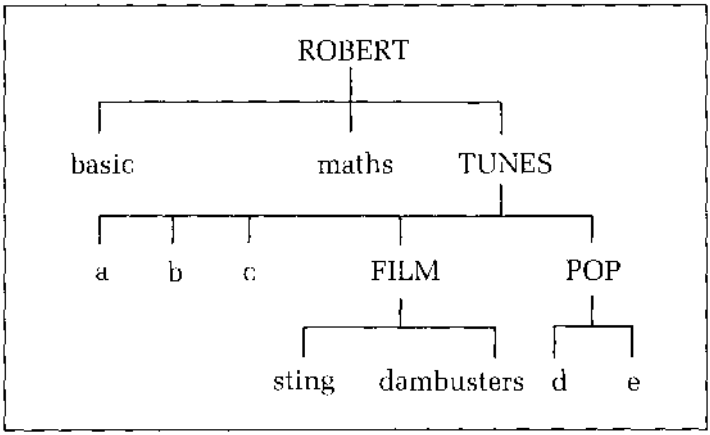
You *now*

type: *CAT TUNES.FILM[RETURN]

and your film-tunes directory appears, with two files in it, DAMBUSTERS and STING.

```
Screen: FILM          (014)     Owner
        Master-disc             Option 00 (Off)
        Dir. ROBERT             Lib. LIBRARY
        DAMBUSTERS WR/WR STING WR/WR
```

You can think of ROBERT's files and directories as a family tree, like this:



In this diagram, files are shown in lower case letters (maths, a, b), and directories in upper case letters (ROBERT, TUNES, FILM).

Pathnames specify particular files or directories by specifying the path down the tree that leads to the file or directory. Each step along the path is a step to the next layer down, and is marked in the pathname by a full stop.

Getting information about sub-directories *EX

*EX produces a display of information about the items in your main directory; you can get displays of information on your sub-directories by

typing ***EX <directory name>[RETURN]**

Selecting directories *DIR

You may want to work in a directory other than the one selected for you when you logged on.

Type: ***DIR <directory name>[RETURN]**

This directory now becomes your currently selected directory. Typing *CAT will produce a catalogue of this directory, not of your main directory.

EXAMPLE
Typing: *DIR TUNES[RETURN]

makes TUNES the *new* selected directory. To load the file STING, you now *need* only to
type: LOAD"FILM.STING"[RETURN]

To return to your main directory

type: ***DIR[RETURN]**

Moving files between directories * RENAME

To move a file from one directory to another

type: ***RENAME <current><new>[RETURN]**

<current> here stands for current pathname;
<new> for new pathname.

EXAMPLES

*RENAME ONE.PROG TWO.PROG

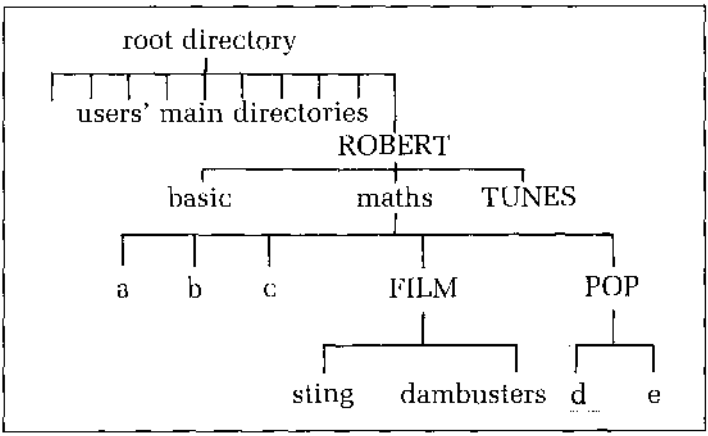
will move the file PROG from directory ONE to directory TWO.

You can change the *filename while* moving it. For example:
*RENAME ONE.JOES-PROG TWO.MY-PROG

will move the file JOES-PROG from the directory ONE to directory TWO, and rename it there as MY-PROG.

The root directory

Outside all the main directories on your disc — and containing them — is the root directory. It's so called because it's at the root of the tree of directories, sub-directories and files.



Displaying the root directory *CAT $

The root directory is called $. So, to display the root directory list on your screen
type: ***CAT $[RETURN]**

EXAMPLE

You have logged on as JULIE.

Type: *CAT $[RETURN]

Screen: $ (009) Public
Master-disc Option 00 (Off)
Dir. JULIE Lib. LIBRARY
BOOT DL/ JULIE DL/
LIBRARY DL/ PASSWORDS /
ROBERT D/ WELCOME DL/

Using other users' files

You can now specify any file in the directories listed in the root directory. You do this by giving the file's full pathname. A full pathname consists of:

- the root directory name ($), followed by a full stop
- the main directory to which the file belongs, followed by a full stop
- the file's pathname.

EXAMPLES

To load ROBERT's file STING

type:

LOAD" $.ROBERT.TUNES.FILM.STING" [RETURN]

Alternatively, you could select ROBERT's directory FILM as your current directory by

typing *DIR $.ROBERT.TUNES.FILM[RETURN]

Then you need only

type LOAD"STING"[RETURN]

Access to others' files

You will be able to load others' files only if their access strings allow you to. Your access rights to their files depend on how they have specified the right hand part of their access strings.

EXAMPLE

You could load someone else's file FILEA whose access string is WR/R, but if you tried to load their file FILEB whose access string is WR/, you would get an " Insufficient access" message.

The owner of a directory — the person with owner access to it — owns all the files and directories in it. Directory owners can:

- set access strings for the files in the directory
- use SAVE and *CDIR to create new files and new sub-directories in the directory
- *DELETE files and directories in the directory.

Other users, with only public access to that directory, can do none of these.

You have owner access to your own main directory and sub-directories, but public access to any directory that you specify beginning with a $ or that you reach (using *DIR) via the root directory.

EXAMPLE

You log on as JOEY. You will have owner access to the files and sub-directories in the main directory JOEY. You will have public access to the contents of the main directory PHILIP, which you have to specify as $.PHILIP. If you specify your own main directory as $.JOEY, instead of just JOEY, you will have only public access to it.

Working with groups of files

Wildcards

It is sometimes useful to refer to a group of files or directories in one command. You can do this by using wildcards — symbols that refer to any character or set of characters in a filename. The two wildcard symbols are # and *:

- # stands for any character
- * stands for any string of characters.

EXAMPLES

AB# *can refer* to ABC and ABZ; it will also match AB, but not ABCD

can refer to any file or directory in the currently *selected* directory

ONE.W* refers to all the files and directories *in* the directory ONE that start with W, including a file or directory called W

When you use a wildcard to refer to a group of files, the reference you give — AB#, for example — is called an ambiguous reference.

Protecting groups of files *ACCESS

You can use ambiguous references with *ACCESS to set access strings for more than one file.

Type: ***ACCESS <reference><access>[RETURN]**

<reference> here stands for ambiguous file reference; **<access>** stands for access string.

EXAMPLES

*ACCESS A### R/R

will give the access string R/R to all your files that start with an A and are up to four characters long.

ACCESS TUNES. L/

will give the access string L/ to all the files in your sub-directory TUNES.

ACCESS Z WR/WR

will give the access string WR/WR to all your files that start with a *Z*, however long those names are, and including a name that is just Z.

Other uses for wildcards

You can use wildcards with other commands, but the effect will be different. The command will not be carried out on all the files and directories to which the ambiguous reference refers; it will be carried out on whichever of those items has the name that comes first in the alphabet. This is because it's impossible to carry out these commands on more than one file at once: the system could not, for example, load two programs simultaneously.
Type: **<command> <reference> [RETURN]**

EXAMPLES

To load *whichever file* comes first in your directory listing
type: LOAD"*"[RETURN]

To get *INFO on the first file that starts with an H

type: *INFO H* [RETURN]

To display a catalogue of the first of your sub-directories
type: *CAT *[RETURN]

Using other discs

What happens when you log on

Most Econet file servers have two disc units, so that two floppy discs are in use all the time. Each time you log on, the file server selects one of these two as your logged disc. It does this by:

- looking at the disc in its first disc unit for a directory that has the same name as your identifier
- if it finds one, it logs you on to that disc
- if not, it tries the other disc
- if it still fails to find a matching directory, it logs you on to the first disc, and selects $ as your directory; you will be given only public access to $.

From then on, the file server uses your logged disc for saving and loading. But you can, during an Econet session, switch to the other disc, if you want to use files stored there. This section describes how.

Selecting another disc *SDISC and *DISCS

To log on to the disc in your file server's other disc drive, you use the command *SDISC.

Type: ***SDISC <disc name>[RETURN]**

Your network manager will tell you the name of the other disc, so that you can use the *SDISC command. Alternatively, you could find out the disc's name by using the command *DISCS.

Type: ***DISCS[RETURN]**

EXAMPLE

Type: *DISCS [RETURN]

Screen: drive disc name
0 Master-disc
1 JMB

Every disc used on your file server is an independent filing system, with its own root directory and tree of directories and files.

When you switch discs, the file server logs you on to the other disc. Unless you have a main directory on both discs, it will select the new disc's root directory $ as your current directory. You will have only public access to it.

Using files from the other disc

You can use a directory or file from the other disc without logging on to it.

You do this by specifying the disc name when giving a command. Type the disc name before you give the name of the file or directory you want.

Type: **<command> :<dn>.<pn>[RETURN]**

<dn> here stands for disc name;
<pn> stands for pathname.

EXAMPLES

*CAT :DISC2.$
*DIR :MASTER-DISC.$.ROBERT.TUNES
*INFO :WORKDISC.$.JULIE.A

When specifying files using this method, the $. is optional.

Machine-code and data files

Saving areas of memory *SAVE

This command is used to save files such as machine-code programs and data. The *SAVE command saves an area of memory as a file.

Type: ***SAVE <filename><s> +<l>**[R**ETURN]**

<s> here stands for start address;
<1> for length of file.

EXAMPLE

Type: *SAVE DATA 3000 +500[RETURN]

This *saves* the area of memory from 3000 to 3500 (hexadecimal numbers) as a file *called* DATA. In this case 3000 was the start address of the area in memory and 500 its length (in bytes).

Alternatively

type: ***SAVE <filename><s><e>[RETURN]**

<e> stands for end address.

EXAMPLE

Type: *SAVE DATA 3000 3500[RETURN]

where 3000 is the start address and 3500 is the end address of the section of memory you wish to save.

You can also specify an execution address. This is the location in the file at which the file server will start when executing the file.

Type:

***SAVE <filename><s> +<l><e>[RETURN] <e>**

stands for execution address.

EXAMPLE

Type: `*SAVE PROG 3000 +300 3030[RETURN]`

This saves a machine-code file which will be executed at `3030` if you load it using * or *RUN (these commands will be explained shortly).

If you leave out the execution address, execution will begin at the start address.

You can also specify a reload address, the location at which the file server will start when it `*LOADs` the file.

Type:

***SAVE <filename><s> +<1><e><r>[RETURN] <r>**

stands for reload address.

EXAMPLE

Type:

`*SAVE PROG 3000 +500 4030 4000[RETURN]`

This saves a machine-code program from an area of memory at `3000` to `3500,` which will be loaded back to the address 4000 by *LOAD, * or *RUN.

If you leave out the reload address, the system will use the start address as the reload address.

Loading an area of memory * LOAD

This command loads files, usually ones which have been saved, using the `*SAVE` command to a particular place in memory.
Type: ***LOAD <filename>[RETURN]**

EXAMPLE

Type: `*LOAD DATA[RETURN]`

to load DATA to the reload address of the file, which is set by *SAVE.

You can also specify where in memory you wish to load a file.

Type: ***LOAD <filename><start>[RETURN]**

EXAMPLE

Type: *LOAD DATA 5000[RETURN]

This ignores the reload address of DATA and loads *the file* at address 5000 *in* memory.

Running a machine-code program * and *RUN

To start a machine-code program, you load the file containing the program to an appropriate address in memory and then begin execution at the execution address of the program. The * command does both these steps by loading a file to its reload address, and then starting execution at the execution address of the file. Both these values are set by *SAVE.

Type: ***<filename>[RETURN]**

EXAMPLE

To run PROG as a machine-code file

type: *PROG[RETURN]

The file PROG will then be loaded at the reload address, and execution will start at the execution address.

In this way you can create machine-code programs and use them as utility commands.

EXAMPLE

If you had developed an editor stored in a *file* called EDIT

typing: *EDIT[RETURN]

would load and *run* the program.

With the *RUN command you can develop command programs which have the same names as Econet filing system commands.

Type: ***RUN <filename>[RETURN]**

EXAMPLE

If you *have* a program CAT which you wish to load and run as a command, typing *CAT *will* display a list of your files.

Type: *RUN CAT[RETURN]

to load and run the file CAT.

Libraries

When you use the *<filename> command — for example, *PROG — the file server will:

- try to find it in your currently selected directory. If the file is not there, it will
- look for it in a special directory called a library.

A library is a directory which normally contains only machine code programs. If the file server finds the file PROG in your library, it will *RUN the PROG program. If PROG is not in the library, you will get a " Bad command" error message.

You can choose any directory as your current library. To check which directory is currently selected

type: ***CAT[RETURN]**

and you will see its name after the word "Lib." at the top of your listing.

Many file server discs have a built-in library called $. LIBRARY. If $.LIBRARY is on your disc, the file server will choose it as your library when you log on. $. LIBRARY contains many useful programs —*DISCS, for example.

You will probably find that your network manager uses $.LIBRARY to store useful programs, so that all the users on your network can call them up easily.

Selecting libraries *LIB

You can change your currently selected library using *LIB, just as you can use *DIR to change your currently selected directory.
Type: ***LIB <library name>[RETURN]**

EXAMPLE

You have several machine-code programs, which you have filed *in* a directory called PROGS. *When* you *want* to use them, you switch your currently selected library to PROGS.
Type: *LIB PROGS[RETURN]

Recording keystrokes in files *SPOOL and *EXEC

The *SPOOL command stores in a file everything you type in while the file is open — that is, it creates a file that records all your keystrokes. You open the file by

typing: ***SPOOL <filename>[RETURN]**

The file you have opened will contain everything that's displayed on your screen from now until you close it. Close it by

typing: ***SPOOL[RETURN]**

The *EXEC command reads a file you specify as if its contents were being typed in at the keyboard.

Type: ***EXEC <filename>[RETURN]**

One use of *SPOOL and *EXEC is to save you repeatedly typing the same sequence of commands. You:

- open a *SPOOL file
- type in the sequence of commands
- close the * SPOOL file
- *EXEC the file whenever you want to use the sequence.

Commands like *CAT produce confusing results in spool files, because the file server will display a catalogue when you type *CAT; the spool file will then include the catalogue display. To avoid this effect, create your command file as a BASIC program, like this:

```
10 *SPOOL STARTUP
20 PRINT"*CAT"
30 PRINT"*SDISC USER-DISC"
40 PRINT"*LIB MYLIB"
50 *SPOOL
60 END
```

To execute this spool file

type: ***EXEC STARTUP[RETURN]**

NOTE: if your computer is fitted with a disc filing system interface, you could use the DFS commands *BUILD, *TYPE, *LIST and *DUMP to prepare files that contain sequences of commands. Instructions on using these commands are in the BBC Microcomputer Disc System User Guide; your network manager should have a copy.

Setting up automatic start routines

Autostart at log-on *OPT4

You can make your next Econet session, and every subsequent session, start automatically with the same action or sequence of actions. To do this:

1 Create a file called !BOOT in your main directory. Put in it the commands you want the file server to carry out each time you log on. To create the file, you use *SAVE or *SPOOL (or, if it's available, the disc filing system command *BUILD).

2 Change the autostart setting of your main directory by
typing: ***OPT4,<number>[RETURN]**

The number you type can be 0, 1, 2 or 3.

0 switches the autostart off
1 makes the file server *LOAD your file !BOOT each time you log on
2 makes it *RUN your file !BOOT each time you log on
3 makes it *EXEC your file !BOOT each time you log on.

Type: ***CAT[RETURN]**

and you will see the autostart option you selected after the word "Option" at the top of your listing.

Each time you log on, the file server:

- finds your directory (if you have one)
- takes the action determined by its autostart option.

If the option is 1, 2 or 3 and there is no file !BOOT in your main directory, you will get the message "Not found".

Once the filing system has finished carrying out its autostart routine, you can continue your session as normal.

EXAMPLE

You might set up a !BOOT file that, *whenever* you log on:

- prints out "Hello"
- gives you a catalogue
- loads the first program in your list (use a wildcard here).

Reading and writing to files

The idea of random access

One advantage of storing your work on discs, rather than on a cassette, is that it is much easier for the computer to find quickly a particular item — just as it's much easier to move straight to a particular track on a record than it is on a cassette. Tape storage systems are called serial-access devices: they have to go right through the tape serially (in sequence) to find what they are looking for. Disc systems, such as your Econet file server, can move straight to the item they want, and are called random-access devices.
As well as making saving and loading operations faster, this random-access feature allows you to select a particular section within a file, and read or write to it immediately. The process is called random-access reading and writing.

Opening a file

To use random access to a file, you:

- say which file you are interested in
- say whether you want to read or write to it or both
- ask the filing system to give you a channel number, by which you will communicate with the file.

All three operations are carried out in one command line.

EXAMPLE
X=OPENOUT("CALENDAR")

This means:

- create and open a file *called* CALENDAR
- open it for both reading and writing
- give it a channel number, and let the variable X stand for that channel number.

The BASIC keywords which ask the file server to open a file for random access are:

- in BASIC I, OPENOUT (to create a new file to write to and read) and OPENIN (to read and update an existing file)
- in BASIC II, OPENOUT (to create a new file to write to and read), OPENIN (to read an existing file) and OPENUP (to read and update an existing file).

BASIC I and BASIC II are two different versions of BBC BASIC. To find out which version you have, look at the notes on software versions in the reference section. The examples that follow use the BASIC II keywords.

You can have up to five channels, so you can get random access to up to five files at any one time. You can open the same file more than once — but only for reading (that is, with the BASIC II keyword OPENIN).

EXAMPLES

```
X=OPENIN("DATA")
```

This means:

- open the file DATA for reading
- let the variable X stand for the channel number the file server assigns to DATA.

```
Y= OPENUP("INDEX")
```

This means:

- open the file INDEX for reading and writing
- let the variable Y stand for the channel number the file server assigns to INDEX.

NOTE: there is no keyword in BASIC I that opens a file for reading only: you will need to use the operating system routine OSFIND. This method is described in the Econet Advanced user guide. Your network manager should have copies.

Using the pointer

The next step is to point to the particular byte in the file that you want to read or write to. The pointer keyword is PTR#.

EXAMPLE

PTR#X=1000

This means: in the file with channel X, point to byte number 1000. The byte number will be increased by 1 each time a byte is read or written. *When* a file is opened, the pointer points to byte 0.

Reading and writing

Now you can read or write to the contents of the byte you have pointed to. The keywords are BGET# (to read a byte) and BPUT# (to write a byte).

E X A M P L E S

NEXTLETTER%=BGET#X

This means: into the variable NEXTLETTER% put the contents of the byte currently pointed to *in* the file with channel X.

BPUT#X,32

This means: into the byte currently pointed to in the file with channel X, put the number 32.

When you are writing to a file, you can increase its size by pointing beyond the current last byte. For example, if you OPENUP a file 20 bytes long and then point to byte 100, using PTR#, the file will be extended to 100 bytes. The extra bytes inserted will be zeros.

Closing the file

When you have finished with your file, close it using the keyword CLOSE#.

EXAMPLES

CLOSE#X

means: close channel X; I have finished with the file.

CLOSE#0

means: close all my open *files.*

Other random access operations

You can use the keyword EXT# to find out how many bytes have been written to the file.

EXAMPLE

SIZE=EXT#X

This means: into the variable SIZE put the size, in bytes, of the file with channel X.

The keyword EOF# is used to tell whether the end of the file has been reached. It is set to true (-1) if the end has been reached, or false (0) if it hasn't.

EXAMPLE

```
10 X=OPENIN("DATA")
20 REPEAT
30 A=BGET#X
40 UNTIL EOF#X
50 CLOSE#X
```

This means: continue to read bytes from the file with channel X — in this example, the file DATA — until the last byte has been read.

There are two more keywords which read and write data to an open file. INPUT# reads data; PRINT# writes data. These keywords are used to read and write BASIC variables, rather than single bytes.

EXAMPLES

INPUT#X, WORD$

reads data from the file with channel X, starting at the byte pointed to, and puts the data read into the variable WORD$.

PRINT#X,"MONDAY"

writes the string MONDAY to the *file* with channel X, starting at the byte pointed to.

There is a summary of all the random access keywords at the back of this guide.

Interlocks

The Econet filing system has a system of interlocks to prevent two users performing conflicting random access operations on the same file at the same time. The system is called multiple reader, single writer; it has these effects:

- once you have opened a file for reading (using OPENIN in BASIC II), anyone — including you — can open it or re-open it for reading but no-one can open it for writing
- once you have opened a file for writing (using OPENIN or OPENOUT in BASIC I, or OPENUP or OPENOUT in BASIC II) no-one — including you — can open it, load it or save it.

If the system stops you opening a file, you will get an error:

Already open

Reading and writing groups of bytes

There is another way of reading and writing bytes, in which you handle them in groups rather than one by one. This method is faster, but more complicated. It uses the operating system routine OSGBPB, which is described in the Econet Advanced user guide.

Printing

If you have a printer server in your Econet you can print out your work as if you had a printer attached to your machine.

Type: ***PS[RETURN]**

You may have to specify the printer server you want to use. You will have to do this if your network has more than one printer server, or if your printer server has a different station number from the one Econet expects. You specify a printer server by typing in, straight after *PS, the station number of the printer server you want to select. Your network manager will tell you what the correct station number is.

Type: ***PS <printer server number>[RETURN]**

The first time you use the printer server after switching on or resetting your station, the computer needs to know that you want to send your work to the printer server, not a printer attached directly to your station.

Type: ***FX5,4[RETURN]**

to use the printer server. You will have to do this again if you reset your machine.
When you are ready to print out your work

press: **[CTRL]B**

or type: **VDU 2[RETURN]**

The network manager may have set up the printer server to print some special text at the beginning and end of each user's work. This is called header and footer text.

The printer will print any header text, followed by anything written to your screen after you pressed [CTRL]B.

If someone else on the network is already using the printer you will see the message "Not Listening" after 30 seconds. If this happens, wait and try again or save your work and print it out when the printer is free.

When you have finished printing

press: **[CTRL]C**

or type: **VDU 3[RETURN]**

to print the last few lines of your text and the footer text.

While you are linked to the printer no one else can use it. If you press [CTRL]B and then don't print anything for 30 seconds, or there is a gap of 30 seconds or more in your printing, someone else can take control of the printer by pressing [CTRL]B at their station.

If you then try to print, you will see the "Not Listening" message.

If no one else presses [CTRL]B, you can carry on printing as normal.

Communicating with other users

You can send short messages to other stations by using the *NOTIFY command.

Type: ***NOTIFY <user id><message>[RETURN]**

or: ***NOTIFY <station><message>[RETURN]**

<user id> here stands for user identifier;
<station> stands for stantio number.

EXAMPLE

You are at station 199. To send the message "Hello" to a user whose identifier is JULIE and who is working at station number 123

type: *NOTIFY JULIE HELLO[RETURN]

or: *NOTIFY 123 HELLO[RETURN]

Julie at station 123 will hear a beep, and this will come up on her screen:

-- 199: HELLO --

The "199" tells her the station the message came from.

If someone sends you a message, it will come up on your screen straight away, perhaps while you are in the middle of a piece of work. Once you have read it, delete it using the [DELETE] key, and the work you were doing will not be affected. Do not press [RETURN] until you have deleted the message.

Using other filing systems

You may have other filing systems available to you. The following keywords are used to select the system you want. To select a system, type its keyword, then press [RETURN].

300 baud cassette system	*TAPE3
1200 baud cassette system	***TAPE12** or *TAPE
prestel and teletext system	***TELESOFT**
cartridge ROM system	***ROM**
disc filing system	*DISC or ***DISK**

To return to the Econet filing system

type: ***NET[RETURN]**

Reference section

Software versions

Some features of Econet and its filing system vary depending on the software your system has. This section describes:

- the two versions of the Econet software
- the two versions of BASIC.

To find out which version of the Econet software you have

type: ***HELP[RETURN]**

A display like this will come up:

NFS 3.34

OS 1.20

In this example, the Econet software (the network filing system or NFS) is version 3.34; the machine operating system is version 1.20.

NFS versions 3.40 and later have the following differences from earlier versions:

- they allow you to prevent your password appearing on the screen, using the procedure described in the Getting going section
- catalogues are displayed in two or more columns
- there are no privileged stations
- there are slight changes in the OSARGS operating system routine.

The concept of privileged stations, and the OSARGS operating system routine, are explained in the Econet Advanced user guide: the network manager should have a copy.

To find out which version of BASIC you have

press: [BREAK]

then

type: **REPORT[RETURN]**

A copyright message will come up on the screen. If the copyright date is 1981, you have BASIC I; if it's 1982, you have BASIC II.

For Econet users, the important difference between the two BASICs is in the keywords used to open files for random access; they are set out in the section on Reading and writing to files.

Commands

This is a list of the commands mentioned in this guide. It covers four kinds of command:

- BASIC commands and keywords (for example, CHAIN)
- operating system commands (for example, *FX5,4)
- Econet commands (for example, *PS)
- file server commands (for example, *ACCESS).

<name> here stands for one of the following:

- filename
- file pathname
- ambiguous file reference
- directory name
- directory pathname
- ambiguous directory reference.

Elements of the syntax shown in square brackets are optional.

* **<name>**
runs the machine-code program specified

***ACCESS <name><owner access string>/<public access string>**
sets an access string for the file or directory

***BYE**
logs the user off

***CAT [<name>]**
displays a catalogue of the directory specified; if no name is specified, a catalogue of the user's main directory is displayed

***CDIR <name>**
creates a directory

CHAIN" <name>"
loads and runs the BASIC program specified

***DELETE <name>**
deletes the file or directory specified

***DIR [<name**>]
selects the directory specified as the user's currently selected directory; if no name is specified, the user's main directory is selected

***DISC** or ***DISK**
selects a disc filing system

***DISCS**
displays the names of the discs currently in use

***EX [<name>]**
displays information on the contents of the directory specified; if no name is specified, information on the contents of the user's main directory is displayed

***EXEC <name>**
reads the contents of the file specified byte by byte

***FX5,4**
selects the network printer

***HELP**
gives information on your software versions

*I AM **[<file server number**>] **<identifier> [<password>]**
logs the user on

***INFO <name>**
displays information on the file specified

***LIB <name>**
selects the directory specified as the user's current library

LIST [<line number>][,][<line number>] lists
all or part of the current BASIC program

LOAD"<name>"
loads the BASIC program specified

***LOAD <name> [<start address>]**
loads the file specified at the address specified

* **NET**
selects the Econet filing system

***NOTIFY <user identifier><message>**
sends the message specified to the user specified

***NOTIFY <station number><message>**
sends the message specified to the station specified

***OPT<number>,<number>**
selects an option; *OPT1,<number> selects a file information display option, and *OPT4,<number> sets an autostart option

***PASS ""<password>**
sets the password specified

***PASS <current password><new password>**
sets a new password

***PS [<printer server number>]**
selects a printer server

***RENAME <current name><new name>**
renames the file specified

REPORT
gives, when entered after [BREAK], the copyright date of the BASIC software in your computer

***ROM**
selects a cartridge ROM filing system

RUN
runs the current BASIC program

***RUN <name>**
runs the machine-code program specified

SAVE"<name>"
saves the current BASIC program as the file specified

***SAVE <name><start> +<length> [<exec>] [<reload>]**
saves the area of memory specified as the file specified

***SAVE <name><start><end> [<exec>] [<reload>]**
saves the area of memory specified as the file specified

***SDISC <disc name>**
logs the user on to the disc specified

***SPOOL [<name>]**
creates a spool file with the name specified; if no name is specified, it ends the current spool file

***TAPE3**
selects a 300 baud cassette filing system

***TAPE12** or ***TAPE**
selects a 1200 baud cassette filing system

***TELESOFT**
selects the prestel and teletext filing system

VDU **<number>**
controls the printer

Command name abbreviations

name	abbreviation	name	abbreviation
*ACCESS	*A.	*BYE	*BY.
*CAT	*.	*CDIR	*CD.
*DELETE	*D.	*EXEC	*E.
*I AM	*I A.	*INFO	*I.
*LOAD	*L.	*OPT	*0.
*PASS	*P.	*RENAME	*REN.
*RUN	*R.	*SAVE	*S.
*SDISC	*SDI.	*SPOOL	*SP.

Random access keywords

In this list:

- **<ch>** means a variable whose value is a channel number
- **<num-var>** means a variable whose value is a number
- **<numeric>** means a number, or a variable whose value is a number
- **<string-var>** means a variable whose value is a string
- **<string>** means a string, or a variable whose value is a string.

<ch>=OPENOUT("<filename>")
creates a file for writing and reading with the name specified

<ch>=OPENIN("<filename>")
opens the file specified for reading and writing (BASIC I only)

<ch>=OPENIN("<filename>")
opens the file specified for reading (BASIC II only)

<ch>=OPENUP("<filename>")
opens the file specified for reading and writing (BASIC II only)

CLOSE#<ch>
closes the channel specified

<num-var>=PTR#<ch>
sets the variable to the number of the byte currently pointed to

PTR#<ch>=<byte number>
moves the pointer to the byte specified

<num-var>=EOF#<ch>
sets the variable to –1 if the end of the file has been reached, otherwise to 0

<num-var>=EXT# <ch>
sets the variable to the size of the file, in bytes

<num-var>=BGET#<ch>
puts into the variable the contents of the byte pointed to

BPUT# <ch>,<numeric>
puts the number or variable into the byte pointed to

INPUT#<ch>,<num-vars or string-vars>
puts data from the file into the numeric or string variables specified

PRINT#<ch>,<numerics or strings>
puts the numbers, strings or variables into the file

Error messages

This is a list of the most common error messages that you may receive at your station. If you get a message that is not listed here, ask the network manager for help.

The numbers shown are the error numbers, in decimal. The error number 168 has several different kinds of error associated with it: this is simply because there are more possible errors than available numbers. You can find out the individual error number of a 168 error by using the operating system routine OSWORD, which is described in the Econet Advanced user guide. There is information on the meaning and use of error numbers in the BBC Microcomputer System *User* Guide, chapter 27; the network manager should have copies.

Already open **194**
You have tried to open a file for random access writing that has already been opened by you or another user.

Bad attribute **207**
You have tried to set an access string that breaks the rules for specifying access strings. Check the section on Protecting your files.

Bad command **254**
You have made a mistake entering a command: the system does not recognise what you have typed in. Try again.

Bad file name **204**
You have tried to give a file a name that breaks the rules for filenames. Check the section on Naming files.

Bad password **185**
Using *PASS, you have broken the rules for choosing passwords. Look at the section on Passwords.

Bad string **253**
The filename or pathname you have specified is too long, or you have omitted a " symbol.

Broken dir **168**
Your directory has been damaged, probably by a disc error. Ask your network manager for help. Individual error number: 66.

Channel **222**
In random access, you have tried to BPUT a byte to or BGET a byte from a file that is not open. This error can also occur if part of your computer's memory — the part that handles its communications with the network — is corrupted. In this case, log on again.

Dir. full **179**
The directory is full: you will have to save files in other directories.

Dir. not empty **180**
You have tried to delete an unlocked directory that is not empty. Delete its contents before deleting the directory.

Disc changed **200**
Check with the network manager.

Disc fault **199**
Tell the network manager.

Disc full **198**
Tell the network manager.

Disc read only **201**
You have tried to write to a write-protected disc: that is, one adapted to prevent its contents from being written over. If you think you should be able to write to the disc, ask the network manager for help.

Entry locked **195**
The file is locked: you cannot SAVE it, *DELETE it, *RENAME it, create a *SPOOL file of the same name, or open it for random access writing. You will have to unlock the file before trying again.

EOF **223**
Stands for end of file. In random access, you have tried to read past the end of the file.

Insufficient access **189**
You have tried to read or write to a file that does not have R or W in the appropriate part of its access string.

Insufficient privilege **186**
Ordinary users are not allowed to do what you have just tried to do.

Is **a dir.** **181**
You have tried to carry out a file operation (for example, LOAD) on a directory.

Line jammed **160**
Tell the network manager.

No clock **163**
Check that your station is plugged into the network. If it is, and you still get the "No clock" message, tell the network manager.

No reply **165**
Tell the network manager.

Not found **214**
The file or directory you specified is not where you specified. Use *CAT to check.

Not listening **162**
If you get this message when you are trying to print some work, the printer is busy. Try again later. If you get the message when trying to log on, or at any other time, tell the network manager.

Not logged on **174**
You are trying to *NOTIFY a user who is not logged on.

Not open for update **193**
In random access, you are trying to write to a file open for reading only.

Outside file **183**
Using OPENIN in BASIC I, you have tried to place a random access pointer past the end of a file.

PW file not found **168**
You may get this message when you try to log on. Tell the network manager. Individual error number: 33.

Rename across discs **176**
You cannot use *RENAME to move a file between directories on different discs.

Too many open files **192**
In random access, you have used all the channels available.

Too many users **184**
You may get this message when you try to log on. Tell the network manager.

Types don't match **175**
You have tried to mix a file and a directory — for example, you have tried to save a file MONDAY over a directory MONDAY.

User not known **188**
You have tried to log on with an identifier the filing system does not recognise. Check what your correct identifier is.

Who are you? **191**
You have tried to enter a command before logging on. Log on.

Wrong password **187**
Check what your correct password is, and try again. If you have forgotten your password, tell the network manager.

Glossary

Access string
a series of letters indicating users' access rights to a file

Catalogue
a list of the contents of a directory

Directory
a collection of files and other directories

Execution address
the location in a file at which execution will start when the file is loaded by the *<filename> command

File
data and programs stored on your file server's discs are organised into files

File server
a computer controlling a disc storage device connected to it .

Identifier
a name by which the network recognises you

Library
directory normally used to store machine-code programs

Load
to transfer information from disc to your computer

Lock
a device to protect a file or directory from deletion: see page 14

Log on
to start a session, specifying your identifier and, if necessary, a file server and a password

Network
a set of computers connected together so that they can exchange information

NFS
network filing system, the software that runs the Econet filing system

OS
operating system, the software that runs your computer's communications with its keyboard, screen and any other devices — a printer, for example — that you might connect to it

Pathname
an extended name for a file or directory that indicates its exact location within the tree of directories and files

Printer server
a computer controlling a printer connected to it

Random access
a feature of disc storage devices that allows you to read and write to particular sections of files: see page 39

Reload address
the location in memory where a file will be loaded by the *LOAD command

Root directory
the directory at the root of — and containing — the tree of directories and files on a disc: see page 24

Save
to transfer information from your computer to a disc

Station
a computer on a network

Wildcard
a symbol that refers to any symbol or group of symbols in a filename: see page 27

Index

Acorn Computers Limited, Fulbourn Road, Cherry Hinton, Cambridge CB1 4JN, England

Printed by Saunders & Williams (Printers) Ltd, Croydon, Surrey