

SJ Research
Hard Disc File Server System Manager's Manual
Release 0.22 dated 17-7-85

This contents section is sent out with each new release of documentation. Please put it in your file along with any new sheets/chapters sent, and throw away the old contents section. If this sheet does not correspond with the contents of your manual, please contact SJ Research.

Copyright
©Philip Spence-Jones & Associates Ltd
108 Mill Road
Cambridge
CB1 2BD

Telephone (0223) 69927.

Contents

1.	How to use this Manual	1-1	11-2-85
1.1	Introduction to the S3 Research File Server Systems	1-2	11-2-85
2.	The System Manager's Basic Guide	2-1	10-6-85
2.0	Starting up with the File Server	2-1	10-6-85
2.1.1	Switching on the File Server	2-1	10-6-85
2.1.2	Switching off the File Server	2-1	10-6-85
2.2	Basic Principles of System Management	2-1	10-6-85
2.3	Keeping a List of Users	2-2	10-6-85
2.4	Entering Users on to the System	2-2	10-6-85
2.4.1	Editing the Password File	2-3	10-6-85
2.4.2	Keeping a Security Copy of the Password File	2-4	10-6-85
2.4.3	Password Files on More Than One Disc	2-4	10-6-85
2.4.4	Creating User Root Directories	2-5	10-6-85
2.4.5	Setting the Account Numbers of URDs	2-5	10-6-85
2.5	Distinction between Users' Names and URDs	2-6	10-6-85
2.6	Crediting or Debiting an Account	2-6	10-6-85
2.7	Other Administrative Operations	2-7	10-6-85
2.7.1	Taking Backups	2-7	10-6-85
2.7.2	Restoring from a Backup	2-8	10-6-85
2.7.3	Copying between Local Disc and File Server	2-9	10-6-85
2.7.4	Multiple Copying between File Servers	2-9	10-6-85
2.7.5	Clearing out Unwanted Directories	2-9	10-6-85
2.8	The Printer Server	2-10	10-6-85
2.8.1	Setting up the Printers	2-10	10-6-85
2.8.2	The Banner File	2-11	10-6-85
3.	File Server System Manager's Reference Guide	3-1	10-6-85
3.1.1	Installation of the Hard Disc File Server	3-2	10-6-85
3.1.2	Connection of Printers	3-2	10-6-85
3.2.1	Operating Instructions for the HDFS	3-5	10-6-85
3.2.2	Indicator Lamps	3-5	10-6-85
3.3	File Server Administration	3-7	10-6-85
3.3.1	Entering Configuration Mode	3-7	10-6-85
3.3.2	Configuration Mode Commands	3-8	10-6-85
3.3.3	Resetting the HDFS Station Number to 254	3-9	10-6-85
3.3.4	Backing-up the Hard Disc on to Tape	3-10	10-6-85
3.3.5	Recovering the System or Data from Tape	3-11	10-6-85
3.3.6	Maintenance of the Tape Drive	3-12	10-6-85
3.4	General Operating Recommendations	3-13	10-6-85
3.4.1	Network Security	3-13	10-6-85
3.4.1.1	Accidents	3-13	10-6-85
3.4.1.2	External Accidents	3-14	10-6-85
3.4.1.3	Deliberate Interference	3-14	10-6-85
3.4.1.4	Special Techniques	3-15	10-6-85
3.5	Command Specifications	3-15	10-6-85
	*CREDIT	3-18	10-6-85
	*DEBIT	3-19	10-6-85
	EDITPASS	3-20	10-6-85
	EDITPRINT	3-23	10-6-85
	*FAST	3-26	10-6-85
	*FLUSH	3-27	10-6-85
	*LOGOFF	3-28	10-6-85
	MONSWITCH	3-29	10-6-85
	*NETMON	3-30	10-6-85
	SETTIME	3-31	10-6-85

	SIZER	3-32	10-6-85
3.6	Spec. for Printer Server Banner and Endtext File	3-33	10-6-85
3.6.1	Special Identifiers	3-33	10-6-85
3.6.1.1	Time and Date	3-34	10-6-85
3.6.1.2	User, Station etc	3-35	10-6-85
3.6.2	Creating the Banner File	3-35	10-6-85
3.6.3	Examples	3-36	10-6-85
4.	Econet Error Messages	4-1	17-7-85
4.1	Network Related Error Messages Generated Locally	4-1	17-7-85
4.2	File Server Errors reported by the BBC Micro as Error A8	4-2	17-7-85
4.3	File Server Errors reported with correct Error Numbers	4-4	17-7-85
4.4	Fault-Finding an Econet Network	4-9	17-7-85
5.	Basic Introduction to Econet for a User		(not issued)
6.	Econet Client Commands	6-1	17-7-85
6.1	Syntax Definitions	6-2	17-7-85
6.2	File, Directory and General Specifiers	6-3	17-7-85
6.3	Wild Card Specifiers	6-5	17-7-85
6.4	Disc Names	6-5	17-7-85
6.5	Printing through the Network	6-6	17-7-85
6.5.1	Selecting a Network Printer	6-6	17-7-85
6.5.2	Turning the Printer on and off	6-7	17-7-85
6.5.3	Off-line Printing from the File Server	6-8	17-7-85
6.6	Command Specifications	6-9	17-7-85
	*ACCESS	6-11	17-7-85
	*ACCOUNT	6-14	17-7-85
	BPUT#, BGET#, PTR#, EXT#, EOF# ,INPUT#, PRINT#, CLOSE#	6-15	17-7-85
	*BUILD	6-17	17-7-85
	*BYE	6-18	17-7-85
	*CAT	6-19	17-7-85
	*CATALL	6-21	17-7-85
	*CDIR	6-23	17-7-85
	*CLOSE	6-24	17-7-85
	COPIER	6-25	17-7-85
	*CV	6-27	17-7-85
	*DEFACCESS	6-28	17-7-85
	*DELETE	6-29	17-7-85
	*DIR	6-30	17-7-85
	*DISABLE	6-31	17-7-85
	*DISCS	6-32	17-7-85
	*DUMP	6-33	17-7-85
	*ENABLE	6-34	17-7-85
	ERAQ	6-35	17-7-85
	*EX	6-36	17-7-85
	*EXEC	6-38	17-7-85
	*FREE	6-39	17-7-85
	*FSLIST	6-40	17-7-85
	*GO	6-41	17-7-85
	*GTIME	6-42	17-7-85
	*GUSER	6-43	17-7-85
	*I AM	6-44	17-7-85
	*INFO	6-46	17-7-85
	*LIB	6-48	17-7-85
	LOAD	6-49	17-7-85
	*LOAD	6-50	17-7-85
	*LOGON	6-51	17-7-85
	MULTICOPY	6-52	17-7-85

*NOTIFY		6-54	17-7-85
OPENIN		6-55	17-7-85
OPENUP		6-57	17-7-85
OPENOUT		6-59	17-7-85
*OPT1		6-60	17-7-85
*OPT4		6-61	17-7-85
OSCLI		6-62	17-7-85
*PASS		6-63	17-7-85
*PRINT		6-64	17-7-85
*PRINTER		6-65	17-7-85
*PRINTOUT		6-66	17-7-85
*PS		6-68	17-7-85
*PTIME, *PDATE, *PDATE2		6-70	17-7-85
*PUTGET, *PUTGET2		6-71	17-7-85
*RENAME		6-73	17-7-85
*RUN		6-75	17-7-85
SAVE		6-76	17-7-85
*SAVE		6-77	17-7-85
*SDISC		6-79	17-7-85
*SPOOL		6-80	17-7-85
*STATEMENT		6-82	17-7-85
*STATIONS		6-83	17-7-85
*TIME		6-84	17-7-85
*TYPE		6-85	17-7-85
*USERS		6-86	17-7-85
*VERS		6-87	17-7-85
7.	Application Note for Econet Programming on the BBC Micro	7-1	17-7-85
7.1	The NFS ROM	7-2	17-7-85
7.2	OSARGS	7-3	17-7-85
7.3	Printing	7-4	17-7-85
7.3.1	PS status enquiry call	7-4	17-7-85
7.4	File Handling	7-5	17-7-85
7.4.1	OPENIN	7-6	17-7-85
7.4.2	File Names	7-6	17-7-85
7.5	Econet Workspace	7-7	17-7-85
7.5.1	Zero Page Workspace	7-7	17-7-85
7.6	Use of OS Workspace	7-8	17-7-85
7.7	Description of the program NETMON	7-9	17-7-85
7.8	A Basic Guide to Econet Line Protocols	7-11	17-7-85
8.	Econet Machine Code Reference Section		(not issued)
9.			
10.	Appendix		(not issued)

Release Note for File Server Documentation ver 0.22

Release made on 30-9-85

This version is made for the Hard Disc File Server (HDFS). It supersedes HDFS System Manager's Manual version 0.20, dated 11-2-85

Chapters 2 and 3 are re-issued entirely. The main changes are

- Sections 2.8 to 2.8.2 -- completely changed to describe new banner file.
- Section 3.1.2 -- More information added about serial port, baud rate etc.
- Section 3.3.2 -- Slight changes to indicator lamp functions
- Section 3.3.4 -- Slight change to N command in tape menu
- Section 3.5 -- Commands BANNER, ENDTEXT, PRE, POST withdrawn. EDITPRINT changed to include banner file name.
- Sections 3.6 to 3.6.3 (added) -- Printer Server Banner File Spec.

Chapter 4 is re-issued, with a new section on network fault-finding (§4.4)

Chapter 6 is re-issued, with a number of new utilities (see beginning of §6.6 for full list)

Chapter 7 is re-issued with corrections.

Chapters 5, 8, 9 and 10 are not released.

Release Note for File Server Documentation ver 0.21

Release made on 10-6-85

This version is made for the Hard Disc File Server (HDFS). It supersedes HDFS System Manager's Manual version 0.20, dated 11-2-85

Chapters 2 and 3 are re-issued entirely. The main changes are

Sections 2.8 to 2.8.2 -- completely changed to describe new banner file.

Section 3.1.2 -- More information added about serial port, baud rate etc.

Section 3.3.2 -- Slight changes to indicator lamp functions

Section 3.3.4 -- Slight change to N command in tape menu

Section 3.5 -- Commands BANNER, ENDTEXT, PRE, POST withdrawn. EDITPRINT changed to include banner file name.

Sections 3.6 to 3.6.3 (added) -- Printer Server Banner File Spec.

Chapter 4 is re-issued, with three new error numbers added. These are

Error 09 - Printer busy with station xxx

Error 0A - Not authorised to use printer

Error C9 - Disc read only

Chapter 6 is identical to ver 0.20, except for one addition in this release .

*PRINTOUT new pages 6-52A and 6-52B

Chapter 7 is identical in the HDFS documentation, and is the same as the release of ver 0.20 on 11-2-85.

Chapters 5, 8, 9 and 10 are not released.

1. Introduction to the S3 Research File Server 1.0

How to use this Manual

This manual is split into sections covering different aspects of operation of the File Server, as follows

- Chapter 1 Introduction and Overall Description
- Chapter 2 Basic System Manager's Guide
- Chapter 3 System Manager's Reference Section
- Chapter 4 Error Reference Manual
- Chapter 5 Econet User's Guide
- Chapter 6 Econet Commands Reference Section
- Chapter 7 Utility Programs Reference Section
- Chapter 8 Econet Machine Code Calls Reference Section
- Chapter 9
- Chapter 10 Appendix

If the system has not been installed by a dealer or by SJ Research, the new owner should read the initial part of Chapter 3 -- the installation instructions.

The new owner of a SJ Research File Server system should start by reading Chapter 5 if he has had no experience of the Econet filing system. If however he has owned either an Acorn Level 2 or 3, or an SJ Research system in the past, he will understand the basic concepts, and should start at Chapter 2.

Having gained familiarity with the system supplied, the system manager (as we will call the person in charge of the network and the File Server) should then read Chapter 3, which contains a full description of the File Server functions.

Please note that Chapter 2 is an introductory chapter to the SJ Research File Server systems, and not a full reference guide. The system manager should read Chapter 3 at the earliest opportunity, so that he understands the full implications of everything that he is doing.

Chapter 4 contains a list of all the error messages that the File Server can produce, with a full description of each one, and the probable cause. It is intended for reference rather than essential reading.

To be able to advise users of the network, the system manager will need to be familiar with Chapters 6 and 7, which describe the system as it appears to a network user at a BBC Microcomputer on the system. For users of the machine code network interface, Chapter 8 provides a full reference section.

The rest of this chapter gives an overall description of the SJ Research File Server system.

Terminology

To avoid cumbersome sentence construction, we have sometimes referred to users (and system managers) as 'he'. We wish to apologise to users and system managers who do not wish to be 'he', and hope that they will appreciate readability to be the better part of correctness.

In common with much of the industry, we believe that computing is excessively male dominated, and try to encourage equality in all ways.

Copyright

Whilst copyright in this document is reserved by Philip Spence-Jones & Associates Ltd, permission is hereby given to owners of SJ Research File Servers to make copies of sections of this manual for their own use, and for the use of their pupils or colleagues.

The sale of such copies is expressly forbidden without explicit permission in writing from Philip Spence-Jones & Associates Ltd.

1.1 Introduction to the SJ Research File Server Systems

The SJ Research Hard Disc File Server is designed to work on Econet® local area computer networks, to the specification laid down by Acorn Computers Ltd as "Econet Level 2" but with a number of additional features. The full features of the SJ Research systems are briefly described below:

- Fully random access to files is offered, either one byte at a time, or a specified block of data (up to the entire file) at a time.
- The concept of "fileowners" (with passwords if required) is offered to restrict access to files by unauthorised users.

Fileowners' directories are themselves files (of a special sort) so that a fileowner can have sub-directories within other directories. In addition, the "root" directory of the system shall be accessible to users, allowing them to access (where authorised) files belonging to other fileowners.

- 20 megabytes (equivalent to 50 standard floppy discs) or more is available on-line without the necessity to change discs. This is important if the file server is used for more than one room (or class, etc) at once.
- A multi-processor architecture allows very fast turnaround of filing system operations. One Z80 based system runs the file server program itself, and a second independent one controls the storage devices and cache memory. An 8237 direct memory access controller allows all devices effectively simultaneous access to the cache. In addition some of the time-consuming operations are performed by hardware, giving at least a tenfold increase in speed for these functions.
- The HDFS offers automated backing up of the main disc, on to magnetic tape in a convenient cartridge. This allows important data to be protected, off-site if need be; data can also be interchanged on this medium.
- A system of accounts is provided, to control both the use of disc space by users, and owner access to files. Each fileowner is allocated one or more accounts, and each account is given a " balance" by the person in charge of the system (the System Manager). Account numbers range between 0 and FF (hexadecimal), although the system manager could of course use 0 to 99 if this is simpler. By allocation of a suitable balance to each account, users can be encouraged to remove unwanted files.
- The account number given to a file controls the owner access to the file, that is to say, the user with access to the account number of a file may read, write or delete that file; if the file is a directory, he may create new files in that directory.
- Each file has also an auxiliary account number. This can be set by the fileowner. The user(s) with access to the account with number equal to the file's auxiliary account number, also has owner access to that file.

- If an attempt is made to load or open any file that is not in the currently selected directory, the library directory will be searched for the file. This will occur regardless of the method of access to the file (Level 2 specifies this feature only for * commands). When a user logs-on, the file server will automatically select the directory LIBRARY (if it exists) as library directory.
- A real-time clock is included in the File Server. When the full information about a file is requested, the system will display the date of first creation of the file, and the date and time that the file was last updated. In addition, utility programs (for example *TIME) are provided to read the clock directly. The clock runs from an internal rechargeable battery when the system is switched off.
- A printer server is included in the unit, allowing any station on the network to route its output to a printer, via the network itself. The printer server has two independent output channels. One supports a parallel "Centronics" type interface, the other is a serial RS232 type. Both printers may be connected at once, and may perform separate print jobs simultaneously.

Note that the additional feature of accounts need not affect the ordinary users, unless they specifically wish to make use of them. In fact, the system manager may turn off the accounting system completely where no data security is required. In this case all users would have access to everything on the system. Alternatively, he may keep data security but disable the disc space accounting part of the system, by setting all accounts to a suitably large value. See Section 6.6 (under *ACCOUNT and *ACCESS) for a full description of account control.

2. The System Manager's Basic Guide

2.0 Starting up with the File Server

S3 Research File Servers are designed to be configured easily without the need for much special knowledge. This section explains how the system manager should get going, setting up the communal filing system for all his users.

This chapter is kept deliberately simple, in order to explain the principles. System managers will find full descriptions of all the commands and utility programs in Chapters 3 (system manager's utilities) and 6 (for all users).

If the File Server has not been installed for you, please refer to Section 3.1, which gives full instructions. You will also need a FAST terminal ROM to be installed in one BBC Microcomputer -- this is described in Section 3.1.1.

2.1.1 Switching on the File Server

The Hard Disc File Server needs only to be switched on at the front panel key switch. For normal operation turn the switch to the SECURE position. However, if you want to perform any system privileged operation (for example editing the Password file or setting the internal clock), then turn the switch to the SYST position. The yellow DISC ACTIVE lamp will flash for a period of about 15 seconds, and then the green ON LINE lamp will light to show that the system is ready for operation.

2.1.2 Switching off the File Server

To switch off the File Server, turn the front panel key-switch to the STOP position. After a pause the front panel lights will extinguish, once the system has written back any data from memory. If the green ON LINE lamp flashes, this means that there is still data in the printer buffer(s). The unit will remain on until the printing has finished, but if you wish to jettison this output, turn the front panel switch briefly to SECURE and back to STOP again.

Do not switch off by unplugging the mains, or from a master switch. (If a master switch is in use, then the File Server should run from a supply independent of it). If the system had not written back all the disc information before switch-off, it will probably be necessary to restore the system from a backup, as the data on the disc will be corrupted.

We recommend not switching off the File Server at all, since hard discs are considerably more reliable if they are kept spinning continuously at all times. If your local Fire and Safety regulations permit it, this is the best course.

2.2 Basic Principles of System Management

Your system will have a number of different users, and you will want to be able to give them facilities to create files for themselves, and also to read certain communal files (for example library programs), and to have selective access to other users' files.

The list of authorised users in a SJ Research File Server is kept in a file called the password file. This file can be read and saved only by someone with system privilege -- normally only the system manager himself and only when the front panel key-switch is turned to the SYST position. The password file contains information

about each user. his password, any accounts he has access to, and administrative information concerning start-up (boot) options, library directories and user root directories.

If someone logs-on to the system, and his name does not appear in the password file, then he will be logged on as the default user, if one has been set up by the system manager using EDITPASS (q.v. §3.5). If no default user has been set by the system manager, the user will receive the message **User not known**.

When a user listed in the password file logs-on, he will be given any rights and privileges listed against his name in the password file. At the same time, any password he quotes will be checked against the one in the password file, before the log-on is allowed to proceed. The system will then search the disc on which the user's password file entry was found, for a directory with the same name as the user's name, and will set this to be the User Root Directory (see Section 6.6 under *I AM for details)

As described more fully in Section 6.6 (under *ACCESS and *ACCOUNT), the account(s) to which a user is given access, control two things .

First, every file (or directory) has an account number, and if a user has access to this account, then he is an owner of that file (or directory). Only an owner may create files in a directory, and only an owner may delete a file or change its access letters (see Section 6.6 under *ACCESS command). Note that there can be more than one owner of a file (or directory), simply by allocating access to its account to more than one user -- this can be useful for communal files in a project.

Second, each account has a credit balance of storage space, and an attempt to create a file which would cause that balance to become less than zero, will be prevented, and cause the error **Account bankrupt**

2.3 Keeping a List of Users

It is wise to plan your list of users and the accounts for them, on paper, and keep it up to date. There is no security required for account numbers and users' names, and even a moderately sized system can have more users and accounts than can be displayed on a screen.

User names may have up to ten characters, which may include letters, numbers and dashes, and must start with a letter. Normally the user's name would be his own surname or initials. However, user names must be unique in the system, so you may wish to add figures to the end of a name.

Account numbers range between 0 and FF (hexadecimal), but you may of course ignore the hexadecimal part and just use numbers up to 99. Allocating account number 0 gives ownership of the system root directory, so account 0 should be allocated only to system privileged users.

2.4 Entering Users on to the System

Switch on your File Server unit and log-on as a system privileged user. If you have just taken delivery of the system, use the name SYST, with the password SYST, In other words, at a BBC Microcomputer on the network, type

***I AM SYST SYST**

You will then need to edit the password file to enter your list of users and their accounts, then create user root directories for each of them, and then set the account number of each user root directory to the same as that allocated to the user,

so that the user has owner access to this directory. These three operations are described below.

2.4.1 Editing the Password File

Please see Section 2.4.3 if you are likely to require password files on more than one disc (for example -- if you wish some users to have access to the system only when a particular disc is present).

A BASIC program called EDITPASS (fully described in section 3.5 under EDITPASS) is provided in the library of the system. to use it,, check that the front panel key-switch is in the SYST position and type

```
*PROT                (PROT prevents other users peeking your machine)
CHAIN "EDITPASS"
```

The program will ask whether you want to edit the password file from disc (option **E**), or to start a new password file (option **N**). As delivered, the system disc contains a password file with three users SYST, FRED and BOOT, so you would normally edit this one -- it is usually preferable to edit the existing file (unless you have had to delete the password file, in which case use **N**).

The contents of the existing file will now appear, along with a menu of commands. Use command **A** to add new users. To save time, enter all the users to be added at this stage, ending each one by typing a **return**, and end the list by typing **return** on its own.

To allocate accounts, select users one by one, using the brown cursor control keys to move the list past the cursor. After selecting a user, type **X** to enter the expanded mode of EDITPASS. The display will now give the accounts allocated to that user, and will also display other information relating to that user. Press key **A** to enter the account(s) to be allocated to the user. A list separated by spaces, or a series with a dash between numbers, may be entered. To remove accounts, prefix the number(s) with a dash (as minus sign). Press **return** on its own to return to the main list of users, and repeat the account setting for other users.

Before finishing with EDITPASS, check these points:

- a) There must be at least one system privileged user on at least one disc (see Section 2.4.3), otherwise it will not be possible to read or save the password file in the future. If you do not wish to use the name SYST, then enter a suitable name for yourself. Type **X**, give yourself system privilege (type **S**), and allocate yourself all accounts (type **AO-FF**). Give yourself a sensible password that no-one is likely to guess.
- b) There must be a user on at least one disc (see Section 2.4.3) with access to account 0, otherwise it will not be possible to create user directories or edit the password file ! Ensure that you have given yourself (and any other system privileged users) access to account 0. If you omit to do so, you may have to delete the password file and start again -- a tiresome procedure.

The EDITPASS program will display a warning if either (a) or (b) above are not complied with.

- c) It is wise for the system manager to have a different name for day to day use, from the one he uses for system privileged operations. This means that someone watching him log on will probably not see the important password.
- d) If you are going to use the name SYST for the system privileged user, then change the password, otherwise a casual reader of this manual will be able to log on with system privilege.

- e) If you want to have a default user on the system, set this up with option **D** in the expanded mode for the appropriate user. If there is no default user, unrecognised user identifiers will get error **User not known**; if there is, they will be logged on as the default. It is normally recommended that you do not allocate any accounts to the default user, so that someone logging on as an unrecognised user will not have owner access to anything.

When you have completed the editing of the password file, return to the list of users, and press **S** to re-save the file. The program will display the name of the default user (if any) and a warning if there exists no user with system privilege or access to account 0, then will prompt **Save? (Y/N)** -- check that the user list is correct, and that the default user and system privileged user(s) are correct before pressing **Y**.

After using EDITPASS, type *BYE and switch off the BBC Microcomputer. The password file remains in memory after EDITPASS, and could be looked at by an unauthorised user otherwise.

2.4.2 Keeping a Security Copy of the Password File

It is wise to keep a copy of the password file elsewhere, so that the main file can be restored quickly if an accident does happen. An easy way is to use the BASIC program COPIER (described fully in Section 6.6), to copy the file either to a floppy disc on a suitably equipped BBC Microcomputer, or to a file in the File Server. In the latter case, set the access status (see *ACCESS in Section 6.6) of the copy to **PWR/**, and give it an account number that no-one else has access to (e.g. Account 0).

The file itself is called %PASSWORDS, and is not in any particular directory. If a mishap does occur, use COPIER again, copying from the backup copy, back to %PASSWORDS

After using COPIER for this purpose, type ***BYE**, and switch off your computer, since the file will be left in the computer's memory.

2.4.3 Password Files on more than one Disc

On a hard disc system which contains more than one drive, it is possible to maintain different password files on the two discs. The system will search through the password file on every disc, starting at drive 0, until a match is found with the user name.

When a user logs on, the system will search for his name in the password file on each disc, starting from the lowest numbered disc in the system. When it finds this user identifier, it will search the same disc (and no others) for a user root directory. Hence a user's entry in the password file and his root directory should appear on the same disc, otherwise this automatic selection will not take place.

See Section 6.6 under ***I AM** command for full details of the action of logging on. Note in particular that it is not desirable for a particular user to appear in the password file of more than one disc, since the stored password and account information may be different on the various discs, and the version used by the system will be the one in the lowest numbered drive -- this could change at random for the user depending on the order that discs had been inserted.

If any disc in the system does not have a password file on it, the File Server will give system privilege to anyone attempting to log on. As delivered the system will have the standard password file on the lowest-numbered drive, and a null file on all others, so this cannot happen accidentally. If the password file has to be deleted (this is done from Configuration Mode -- see Section 3.3), ensure that the system is

left with at least a null password file %PASSWORDS on every disc (with, of course, a 'real' password file on one or more discs). The null file can be saved with the program EDITPASS -- after selecting the appropriate disc. Log on as a system privileged user (if one exists in any password file at this stage) with the front panel key-switch in the SYST position, type **CHAIN "EDITPASS"**, then **N** for a new file, then **S** to save an empty file.

2.4.4 Creating User Root Directories

You need to have access to account 0 to create directories in the root, as described in 2.4.1 above. In addition, if you are going to allocate accounts to users, you will need access to all other accounts. Check that you are in the disc root directory (that is the directory in which all the user root directories appear) on the desired disc, by typing

```
*DIR $          or  
*DIR $<disc name>
```

Then create a directory for each user, using the CDIR command

```
*CDIR ALLEN  
*CDIR BURTON
```

and so on. At the end, check that the directories exist by typing ***CAT**, which will give a list on the screen of all the directories in the root.

2.4.5 Setting the Account Numbers of User Root Directories

You have already allocated one or more accounts to each user whilst editing the password file. In order for the users to have owner access to their own user root directories, you need to give the corresponding account number to their root directory. The file server will then automatically take care of accounting, by the rule that any file (or sub-directory) created will have the same account number as that of the directory that it is in. A user may change the account number of a file, but only if he has access to another account to change it to.

Before setting account numbers, ensure that each account that you are going to allocate has some credit. See Section 2.6 for information about ***STATEMENT** and ***CREDIT**.

The root directory \$ on each disc always has account number 0. By the above rule, all the directories will initially have account number 0, and hence only users having access to account 0 will own them, allowing them to create files in them.

To change the account numbers of these directories, use the ACCOUNT command. Suppose that you had allocated account 22 to user ALLEN, and account 23 to user BURTON. Then type

```
*ACCOUNT ALLEN 22  
*ACCOUNT BURTON 23
```

and so on. You need to have access to all accounts to be able to do these allocations -- how to do this is explained in Section 2.4.1, particularly in paragraph (a).

You may get the error message **Account Bankrupt** whilst running the ***ACCOUNT** command. It will be necessary credit the account using the ***CREDIT** command (see below).

If you have set a default user in EDITPASS (section 2.4.1 paragraph (e)), you will probably want to set up a user root directory for the default, and put a !BOOT file in it (see Section 6.6 under *OPT4 for details). In this case, it is recommended that you leave this directory with account number 0, so that only you can change it.

Alternatively, you may wish to allocate some 'scratch-pad' filing space to anonymous users, in which case both the default user and the corresponding user root directory should be given an account (with the balance suitably set using *CREDIT) to permit this.

2.5 Distinction between Users' Names and User Root Directories

There is the possibility of confusion between a user called FRED (for example) and a user root directory (or URD) called FRED. The concept of a user exists only in the password file, and is the means of identification for the purpose of allocating accounts, privileges and other options.

At log-on, the system will search the disc on which the user was found in the password file, for a URD of the same name as that of the user, as a convenience to that user and others. (Note that this is different from Acorn systems, which search all discs for a URD). For example, others can gain public access to FRED's files (assuming that he has allowed public access) with comands like

LOAD "\$.FRED.PROGRAMS.EXAMPLE1"

Note however that there is actually no need to allocate a separate URD to every user, nor is it necessary for the URD actually to have the same name as the user. The allocation of accounts is the only thing that determines whether a user has owner access to a file.

For example, for a sixth form project with three people working on it, it may be useful to have a root directory called PROJECT. Suppose the three users had user names JOHN, KEN and PHIL, you might allocate them all account 98. Then set the account number of PROJECT to 98, and all three users have owner access to directory PROJECT, and can all create files in it. At the same time, any or all those users can be given owner access to other files or directories, by allocating them other accounts, so that you could also allocate account 43 to user JOHN, and then set up a URD called JOHN, with account number 43 -- only JOHN would have owner access to this latter directory.

If there is no URD called PHIL on the disc where user PHIL was found in the password file, then when user PHIL logs on, the File Server will select the root directory \$. PHIL will need to select directory PROJECT explicitly before he can begin work, by typing

***DIR \$.PROJECT**

2.6 Crediting or Debiting an Account

On the Hard Disc File Server, all the accounts are pre-set with credit balances which should be useful. To find out what they are, log-on as a user with access to all accounts, and type

***STATEMENT**

This will produce for each disc a list of all the accounts to which a user has access, with the associated credit balances. For the system manager it would be wise to send this information to a printer, so type

***PS** this selects the network printer
<Ctrl-B>*STATEMENT <Ctrl-B> means press B while holding down CTRL key
<Ctrl-C>

The network printer will then produce a copy of the screen. To change the balance of any account, use the DEBIT and CREDIT commands. For example

***CREDIT 43 500**

will add another 500 kilobytes of space for use by files with account number 43. Similarly *DEBIT will subtract space from that allocated. These two commands require system privilege and the front panel key-switch must be in the SYST position.

These commands only change the outstanding credit balance of available space. Since the credit balance cannot exceed 65535 kilobytes, the command

***DEBIT 44 65535**

is guaranteed to wipe out any credit balance left to account 44.

When a file is created, the account corresponding to its account number is debited by an amount equal to the size of the file (but not its extent - see Section 6.6 under OPENOUT). If there is insufficient balance to allow saving of a file, the error message **Account bankrupt** will be produced, and the user must either delete something, or move some files to another account -- assuming that he has access to another.

On a system which has more than one disc in use (i.e. most floppy disc systems and a few of the larger HDFS units), the STATEMENT utility will print a list of the accounts to which the user has access, separately for each disc. The user can however only save files on discs which contain one or more directories with account number(s) to which he has access.

2.7 Other Administrative Operations

The system manager needs to be able to perform various other operations. These include taking tape backups, clearing directories of unwanted materials, copying between cassette tape or disc and the File Server, and managing the Printer Servers built into the File Server system.

2.7.1 Taking Backups

The tape unit on the HDFS has a capacity large enough to back up one 20 megabyte disc on to a single cartridge.

To make a backup, turn the front panel key-switch to the SYST position, and press the ENTER CONFIGURATION MODE switch on the rear panel to the left (as viewed from in front of the HDFS unit). The red CONFIGURATION MODE lamp will flash, and the green ON LINE lamp will extinguish.

On the BBC Microcomputer equipped with the FAST ROM (see Section 3.1), typ

***FAST**

The FAST terminal program will reply with

Station number to attach to:

Type in the station number of the File Server (usually 254 unless you have changed it

-- see Section 3.3.2 or 3.3.3). The BBC Micro is now acting as a terminal to the HDFS, and the following menu will be displayed

SJ Research Hard Disc File Server

- B boot FS**
- C various tests**
- H this help msg**
- N new disc name**
- S set FS station number**
- T tape routines**

A)

(The command C and any others not in the menu are for system maintenance by expert personnel only !) Press **T** followed by **<return>** for any tape command, and the system will respond with

Tape Commands

- N normal operation - i e**
(Format)/Backup/Check tape
- C Check tape against disc**
- F Format tape**
- V Verify tape**
- B Backup data/system**
- R Restore data/system**

Command:

Press **N** for a normal backup -- the system will ask whether the tape is to be formatted (this is only necessary the first time a tape is used), then back up the data and program areas on the disc, on to the tape, and then check the tape to ensure that the data is read back correctly. This entire process takes about 30 minutes (20 minutes if formatting is not required), but you need not be present for this time.

After doing a back-up, press **B<return>** to re-start the File Server program

2.7.2 Restoring from a Backup

If trouble is experienced with a SJ Research File Server system, please get in touch initially either with your dealer, or with SJ Research directly. Sometimes problems occur which appear to be file server problems but are in fact due to other unconnected causes.

The following occurrences can be symptoms of system problems, and you should seek advice if they occur.

A file is obviously corrupted. It is quite likely that only one block (of size 1 kilobyte) will be affected. Check that the file is not likely to have been corrupted by a program that accesses it, or similar cause.

A directory is corrupted. Commonly there will be a repeated name, or names that are obviously rubbish. An attempt to read the affected names may well give a **not found** error.

The disc root directory is corrupted. Usually it will not be possible to gain access to any files on the system.

The File Server system crashes regularly, for no apparent cause.

In all these cases the action is the same. Take a backup of the system as it is, if this is possible. This backup is for safety, in case you lose important data by restoring from a previous version of the data. In addition, if the failure was due to a bug, then SJ Research will need this backup to do a post-mortem (they will often be able to recover any vital data at the same time).

Restore the data from a recently backed-up version. Enter configuration mode as above, then type **T<return>**, followed by **R**.

The system will prompt

D data area
S system
V variable

Press **D** to restore the data area, and/or **S** to restore the system area. The data area will take about 10 minutes to restore, the system area about one minute.

Please note that if you have deleted a file accidentally, it is usually easier to re-enter that file unless this is likely to take more than about half an hour. If you do need to recover a file from tape in the current version of File Server software, first back up the data area on to tape. Next, restore the data area from the appropriate backup tape. Copy the desired file(s) on to floppy disc (or other medium) using the BASIC program COPIER. Then restore the current data area from the tape on which you have just made the backup.

Press **B<return>** to re-start the File Server program.

2.7.3 Copying between Local Disc and File Server

This is done from any BBC Micro on the network which is also fitted with a disc (DFS) interface, and a disc drive. A BASIC program called COPIER is provided as a general purpose copying utility, for files of any length. It is supplied in the library -- to use it type

CHAIN "COPIER"

The program will prompt for the source and destination filing systems. You may specify DISC, NET or other filing system, and use the program to copy either way. You will then be asked for the file names, and the new names. If the new name is to be the same as the name on the original system, then press just **<return>** after the **New name** prompt.

The use of COPIER is not recommended for copying between File Servers, or between directories of one File Server. For this purpose use MULTICOPY (See Section 2.7.4 below)

COPIER is fully documented in Section 6.6

2.7.4 Multiple Copying between File Servers

The BASIC program MULTICOPY can copy entire directory trees between File Servers. To use it, type

CHAIN "MULTICOPY"

The program will prompt for file server number and user name for the source and destination File Servers. It will then request the path name of the directory tree to be copied -- specify \$ if the entire File Server contents are to be copied.

MULTICOPY is fully documented in Section 6.6

2.7.5 Clearing Out Unwanted Directories

The BASIC program ERAQ will erase a complete directory tree. To use it, type

CHAIN "ERAQ"

The program will prompt for the path name of the directory tree to be deleted, and will also ask if you want to OK the files before deletion. For example, to erase all trace of directory FRED, type \$. FRED for the path name. The files in directory FRED and all its sub-directories will be listed and deleted.

ERAQ is fully documented in Section 3.5

2.8 The Printer Server

S3 Research File Servers all contain a built-in printer server with facility to connect one or more printers. When users select the network printer using ***FX5,4** or ***PS**, their output will appear on the printer(s) as set up below. Usually a banner will be printed before their output. This is a text string set up by the system manager, which may also contain information (such as user identifier, time, date etc.) inserted into the string by the system.

An example of a banner is

SJ Research File Server * Station 5 (FRED)**

08feb85 15:21:04 ***

The system can also print a standard string at the end of each user's printout. An example may be a row of asterisks followed by a page throw.

One or two printers may be connected to the Hard Disc File Server. One to the parallel output and one to the serial output. The connection of printers is described fully in Section 3.1.2.

The printer(s) connected to the File Server need to be set up, and given names. Section 6.5 explains printing through the network, and section 3.5 under EDITPRINT gives a full description of how to set up the printers.

The program PS (mentioned above) is supplied in two versions, called **PSFX60** and **PSnoFX6**. If your printer does an automatic line-feed after every carriage return, then use PSnoFX6, otherwise PSFX60. The latter does the call ***FX6,0** which allows line-feed characters through the BBC Micro printer driver. Please rename the appropriate version as PS for the convenience of all your users.

2.8.1 Setting up the Printers

To set up the printers, type

CHAIN "EDITPRINT"

The program will respond with

Edit printer details	- 1
Change priority order	- 2
Select system message channel	- 3
Reset print server & exit	- RETURN

Enter choice :

Choosing **1** from the main menu will give the further prompt :

Which printer (1=parallel, 2=serial or COS routine on RM380Z) 1

then the options menu will be displayed, for example :

Printer 1

- 1 - Name : PARALL**
- 2 - Printer exists : Yes**
- 3 - Anonymous users allowed : Yes**
- 4 - Account ownership required : No**

6 - Banner File Name :

Change which item (1-6, RETURN to save changes) :

The printer can be given a name of up to six characters -- more characters may be entered, but only the first six will be recorded or subsequently checked. The names PRINT and AUTO are reserved and may not be given to a printer (see section 6.5 on printing and section 3.5 under EDITPRINT). This should be repeated for the serial printer. Printers should be set with **No** after **Printer exists**, if they are not connected.

The banner information is contained in a text file called the banner file. The format of the banner file is described below, and the name of the file should be entered after Option 6. Initially no banner file is selected, which means that no banner will printed -- it may be useful to leave no file selected for special devices like plotters.

The printer server can perform an automatic selection of printers in a pre-determined order. This order is set up by option 2 from the main menu. The program will prompt .

This option allows the system manager to select the order in which printers are selected for users who have not requested a particular printer.

Current order :
1st choice is printer 1
2nd choice is printer 0

New 1st choice
(number of printer, or 0 if none) :
New 2nd choice
(number of printer, or 0 if none) :

The File Server will print messages after a system failure, or continuously if the Monitor or Debug outputs are turned on (this is done with the program MONSWITCH (q.v. §3.5)). The printer on which system messages appear is selected by option 3 from the main menu -- the program will prompt :

System messages currently go to printer 1

New destination :

Although the system message channel can be set to 0, system messages after catastrophic errors will still appear at printer 1. This is because the message can be important in these cases. The printer specified can also be used for users' output, since the system messages will be separated by a page throw and header as normal.

If Monitor or Debug are on, however, a lot of output is produced and should go to a printer that has been set as **Non existent** above (this will prevent users from accessing it).

2.8.2 The Banner File

The text string that specifies the banner that is printed at the head of each user's output (and the characters printed after it), is presented to the system in a standard text file. This file can be anywhere in the File Server, subject to the restriction that only 23 characters can be used to specify it to EDITPRINT.

If two printers are connected, they may use the same banner file, or different banner files. The latter option is more useful if the printers are of different types, requiring perhaps different control codes. If no banner file is specified for a printer, then no banner will be printed. This is also the case if the system does not have read access to the banner file (i.e. there is not access letter **R** in the owner category for the file), or if it cannot be found.

A banner file contains text of the form

```
end-text <BANNER> banner text
```

The special word **<BANNER>** must be enclosed in angle brackets as shown. Note that the end-text (the characters to be printed after the user's output) comes first, and the banner text itself after the word **<BANNER>**. The texts will be printed literally until one of the following characters or identifiers is encountered .

	introduces a 'control' character, e.g. A inserts <ctrl-A> ? inserts ASCII character &7F (delete). ! inserts the next character with &80 added (i.e. top bit set). < or > inserts characters < or >.
< USERNAME >	The user originating the print job. No leading spaces.
< STATION >	The station originating the print job. Leading zeroes printed.
< HOURS >	Time-of-the hours, 24 hour clock. Leading zeroes printed.
< MINUTES >	Minutes past the hour. Leading zeroes printed.
< SECONDS >	Seconds past the minute. Leading zeroes printed.
< 12HOURS >	Time-of-the hours, 12 hour clock. Leading zero replaced by space.
< AM >	The word am or pm as appropriate.
< DATE >	The day of the month. Leading zero replaced by space.
< ST >	The correct suffix for <DATE> -- e.g. rd on the 3rd.
< MONTHNAME >	The month, in full. No leading spaces.
< MTH >	The month, three letter abbreviation, all lower case, e.g. jan .
< MONTH >	The number of the month, leading zero printed, e.g. Feb is 02 .
< YEAR >	The last two digits of the year.

Note that all the identifiers are enclosed in angle brackets <>. Unrecognised identifiers will be printed literally. There are more identifiers than in the list above -- a full description of them, and other rules for banner strings is given in Section 3.6.

There are two banners supplied as standard : these are in a directory called BANNERS in the root directory \$. The file \$.BANNERS.SIMPLE contains the following string :

```
|L<BANNER>SJ Research File Server *** Station <STATION> (<USERNAME>)  
<DATE><MTH><YEAR> <HOURS>:<MINUTES>:<SECONDS> ***|M|J
```

which will print the banner shown as an example in Section 2.8, and a single page throw at the end of the user's output. The other banner supplied is called \$.BANNERS.FANCY, and contains the following string :


```

|J|J|J<MARK><USERNAME><TAB 11>***** Print started at <START><H>:<M>:<S>
<DATE><MTH><YEAR> ***** <USERNAME>
|J<MARK><USE.RNAME><TAB 11>***** Print ended at <END><H>:<M>:<S>
<DATE><MTH><YEAR> ***** <USERNAME>
|L<BANNER><MARK><USERNAME><TAB 17><USERNAME><TAB 34><USERNAME><TAB 51>
<USERNAME><TAB 68><USERNAME>
|J***** SJ Research Printer Server *****
|J<MARK>** Output from Station <STATION> (<USERNAME>) at
<12HOURS>:<MINUTES> <AM> on <DATE><ST> <MONTHNAME> 19<YEAR><TAB 77>**
|J*****
|J

```

This will print a banner of the form

```

FRED          FRED          FRED          FRED          FRED
***** SJ Research Printer Server *****
** Output from Station 068 (FRED) at 3:46 PM ON 22nd August 1985          **
*****

```

user's text

```

FRED          ***** Print started at          15:46:27 22aug85 *****          FRED
FRED          ***** Print ended at          15:48:17 22aug85 *****          FRED
followed by a page throw.

```


3. File Server System Manager's Reference Guide

This chapter contains the following sections :

- Installation instructions for the File Server
- Full operating instructions
- Further recommendations (e.g. security considerations)
- Command Specifications for system operations

The system manager taking delivery of a new system should first read section 3.1 in this chapter. which gives installation instructions for the File Server package. This step may of course be omitted if the File Server is being installed by a dealer or by SJ Research.

Chapter 2 gives a step by step guide for running the File Server after it has been installed. It is wise to read it through before trying anything. and then follow the step by step instructions for adding new users to the system. and so on.

This chapter gives full information about all aspects of File Server operation. including a full description of all the system manager's operations. The commands available for, and intended for use by, normal users are documented in chapter 6.

3.1.1 Installation of the Hard Disc File Server

The Hard Disc File Server is normally installed by a dealer, or by SJ Research. However, these instructions are included in case the system manager wants to move the unit.

The unit should **be** installed in a reasonably dust-free environment, preferably in a small room (for example a preparation or store room) near to the main computer classroom(s). A sturdy shelf or work-top of depth at least 500mm (20 inches) is best for the unit -- this includes the depth of the File Server unit and also about 70mm for connectors on the rear. There is a fan on the rear of the unit which draws air in, to be expelled through the front and top of the case. Care should be taken to keep the fan filter unobstructed (a piece of paper falling down the back will be sucked against the fan intake).

It is recommended that wherever possible a special mains socket is fitted to run the HDFS unit. This socket should be unswitched, and placed behind the unit or otherwise out of the way of accidental unplugging. If the computer area has a master switch, the file server supply should be independent of it.

Connect the printer(s) and set up the serial baud rate (if a serial printer is to be used) as described below. Connect the socket marked ECONET to the network with the lead provided. Connect the unit to the mains supply.

A BBC Microcomputer should be available fairly close by, so that operations using Configuration Mode (see section 3.3.1) can be done without too much walking. This computer should be fitted with the FAST EPROM supplied, as follows .

Remove the four screws securing the lid to the case -- there are two at the back and two (with the larger size of head) underneath the keyboard. Lift off the lid. Remove the two or three nuts and bolts securing the keyboard, and pull the keyboard forward (there is no need to unplug it). Insert the EPROM, with the end containing a semicircular depression pointing toward the back of the computer. The BASIC ROM, which will be marked with the numbers PBO1 or PB05 after the type number, should be plugged in to the right of the FAST EPROM. Reassemble the computer, and check that the EPROM works by typing *HELP -- the response should include **Fast terminal ROM ver n.nn.**

3.1.2 Connection of Printers

Two printer ports are supplied on the HDFS, so that a Centronics type parallel printer and a RS232 type serial unit may be connected.

The parallel printer is connected via a 34 way 'insulation displacement' socket (supplied) and matching ribbon cable, to a suitable mating connector (usually a 36 way Amphenol type 57F-30360) for the printer. The connections are given below in Fig 1. At the File Server end, the stripe on the cable should be at the right-hand side (i.e. nearest to the Baud Rate switch).

The serial printer uses the same connector as on a BBC Microcomputer. This plug may unfortunately be inserted either way into the socket -- no damage will be done, but the printer will not function. The socket is the same way up as on the BBC Microcomputer, so if the user subscribes to the convention of marking the top of the connector with a spirit marker, this will also hold good for the HDFS. The serial connections are given in Fig 2, along with the connections to be made to the industry standard D-type connector which will probably be fitted to a serial printer.

Setting the serial baud rate is done on the small slide switch immediately next to the parallel printer connector. The baud rates available are :

A	19200 Baud
B	9600 Baud

C	4800 Baud
D	2400 Baud
E	1200 Baud
F	300 Baud
G	150 Baud
H	75 Baud

The baud rate required for a particular printer may be found from the printer manual (commonly printers are set for 9600 Baud). If the printer baud rate can itself be set, we recommend that 9600 Baud is selected for both.

The components to make up a printer lead may be bought from most computer dealers or from component distributors such as RS Components, or suitable made-up cables can be obtained from SJ Research.

The pin assignments for a CENTRONICS parallel printer interface are outlined below. For detailed information consult your CENTRONICS style printer/computer manual.

NOTES:
 1. "Direction" refers to the direction of signal flow as viewed from a printer.
 2. Pin 34-36 can have special usage on specific models.

Signal pin No.	Return pin No.	Signal	Direction	Description
1	19	STROBE	In	STROBE pulse to read data in. Pulse width must be more than 0.5µs at receiving terminal. The signal level is normally "HIGH"; read-in of data is performed at the "LOW" level of this signal.
2	20	DATA 1	In	These signals represent information of the 1st to 8th bits of parallel data respectively. Each signal is at "HIGH" level when data is logical "1" and "LOW" when logical "0".
3	21	DATA 2	In	
4	22	DATA 3	In	
5	23	DATA 4	In	
6	24	DATA 5	In	
7	25	DATA 6	In	
8	26	DATA 7	In	
9	27	DATA 8	In	
10	28	ACKNLG	Out	Approx. 5µs pulse. "LOW" indicates that data has been received and that the printer is ready to accept other data.
11	29	BUSY	Out	A "HIGH" signal indicates that the printer cannot receive data. The signal becomes "HIGH" in the following cases: 1. During data entry 2. During printing operation 3. In OFF-LINE state 4. During printer error status.
12	-	PE	Out	A "HIGH" signal indicates that the printer is out of paper.
13	-	SLCT	Out	This signal indicates that the printer is in the select state.
14	-	OV	-	-
15	-	NC	-	Not used.
16	-	OV	-	Logic GND level.
17	-	CHASSIS-GND	-	Printer chassis GND. In the printer, the chassis GND and the logic GND are isolated from each other.
18	-	+5V	-	For CENTRONICS use only.
19 to 30	-	GND	-	RETURN signal GND level.
31	30	PRIME	In	When the level of this signal becomes "LOW", the printer controller is reset to its initial state and the print buffer is cleared. This signal is normally at "HIGH" level, and its pulse width must be more than 50µs at the receiving terminal.
32	-	FAULT	Out	The level of this signal becomes "LOW" when the printer is in - 1. PAPER END state 2. OFF-LINE state 3. Error state
33	-	GND	-	Same as with Pin Nos. 19 to 30.
34	-	NC	-	(see note 2)
35	-	NC	-	(see note 2)
36	-	NC	In	(see note 2)

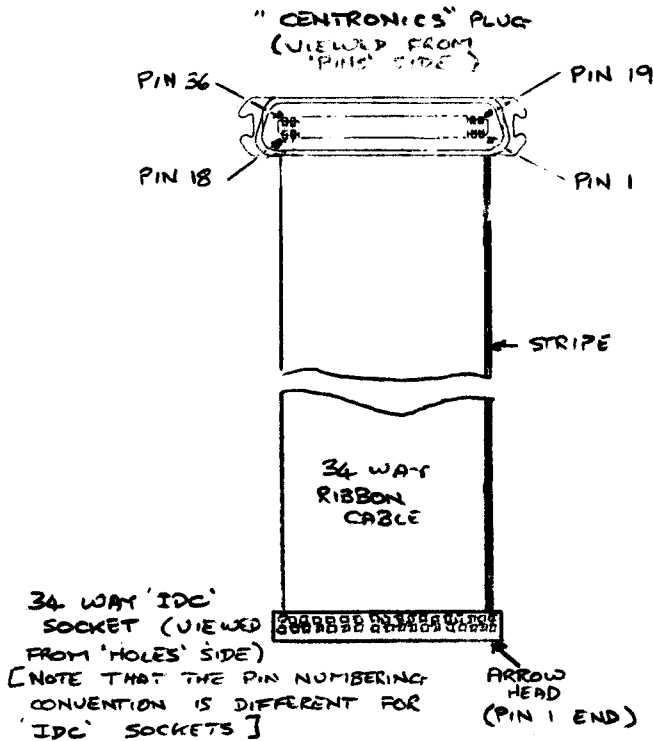


Fig 1 Parallel Printer Connections

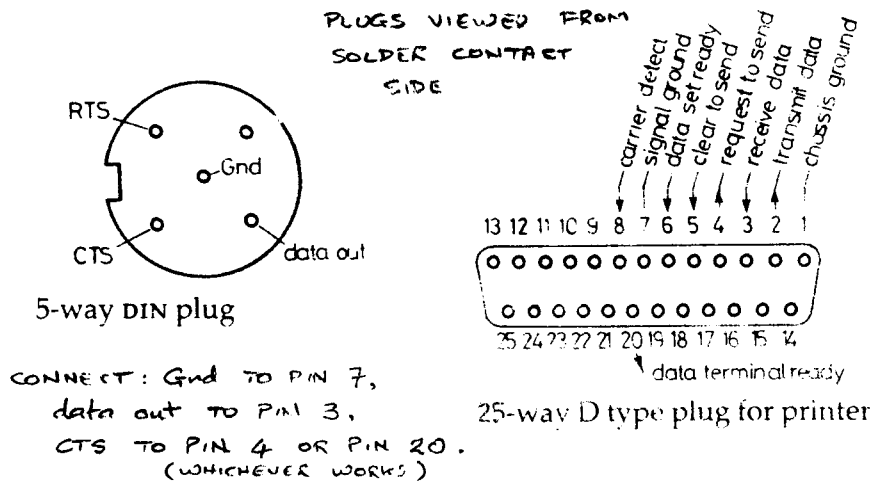


Fig 2 Serial Printer Connections

3.2.1 Operating Instructions for HDFS

The Hard Disc File Server is started simply by turning the front panel key-switch from the STOP to the SECURE or SYST position. After about a second the yellow DISC ACTIVE lamp will light for about 20 seconds, then go out for 5 seconds, and back on for a further period (depending on the quantity of information stored, but usually 30 seconds to 5 minutes). then the green ON LINE lamp will light.

If any other lamps light. this probably indicates a problem with part of the system. See section 3.2.2 below for full information on the lamps.

To switch off the HDFS. turn the key-switch to the STOP position. The unit will run on for a period, until all the information in its memory has been written back to the disc. If there is any printer output in progress, this will also be completed -- the green ON LINE lamp will flash if this is the case.

When all these operations are finished, the power will switch off automatically.

The printer output can be terminated prematurely by turning the key briefly on and off again. The contents of the printer buffers will be thrown away. and the File Server will stop almost immediately.

Note that Hard Disc units are considerably more reliable if they run continuously for their full operational life. Hence it is recommended that the File Server is left running in the SECURE (or the SYST) position all the time. If a hard disc is going to fail, it will generally do so when the disc is gaining speed after switch-on. There are many examples of large hard disc units having run without interruption for 10 years and more as a result of this policy.

If local rules (for example fire regulations) make it unavoidable that you switch off the unit. always switch off by the key-switch; NEVER SWITCH OFF BY UNPLUGGING OR TURNING OFF THE UNIT AT THE WALL SOCKET. OR AT A MASTER SWITCH. If this is done. it is possible that the file server program will not have written back the contents of the cache memory. so that the information on the disc will not be in a consistent state. This could mean that files or directories becomes inaccessible. if the corrupted directory is the root directory, all the information on the disc could be lost. and it will be necessary to recover a previous version from tape, with possible loss of one or more days' work.

3.2.2 Indicator Lamps

This section gives a full description of the meaning of the nine indicator lamps.

CHANGE TAPE (green) indicates when the system has finished with a tape, and tells the user that the tape cartridge may be removed. A tape should not be removed after a tape operation until this lamp lights.

TAPE ACTIVE (yellow) indicates that information is being written to or read from the tape. The tape drive should not be opened while this lamp is lit -- the system manager should wait until the green CHANGE TAPE lamp lights.

DISC ACTIVE (yellow) indicates that the hard disc is being accessed. It may not light if a file is read or loaded for a second or subsequent time. The HDFS has a very large cache (intermediate) memory for data in transit between disc and network, and if the system is not in heavy use, the file will still be in cache (and incidentally will be transmitted almost immediately. without needing to wait for the disc to find it again). If the DISC ACTIVE lamp remains on continuously for more than a few minutes. this indicates a hardware fault.

SYSTEM FAIL (red) indicates that the file server program has ceased to run normally. usually as a result of a disc error or other hardware problem. This lamp will usually flash. and the only way to recover is to turn the front panel keyswitch

to STOP, or to enter Configuration Mode.

A message will be printed on the system messages printer (initially the parallel printer until the system has booted up. thereafter the printer selected by EDITPRINT). If the system manager attempts to switch off or enter Configuration Mode, then the green ON LINE lamp will flash if there is a message and no printer is connected. or if the printer buffer is full of users' output. To discard this output. move the front panel keyswitch to another position and back again (or press the ENTER CONFIGURATION MODE switch again).

NO CLOCK (red) comes on if the Econet clock signal is not present at the network connector on the rear. most commonly the HDFS will have been unplugged from the network. or someone has unplugged the clock box itself. If the clock box is connected and working, unplug it from the network and connect it directly to the File Server unit only, and check that the NO CLOCK lamp goes out. If it does not. there is a hardware fault either in the clock box or in the HDFS or the connecting cable.

BOOT ERROR (red) will light if the system has not managed to load the HDFS software from disc usually as a result of a hardware or disc problem. or if the system manager had restored the File Server system from a faulty or blank tape.

PRINTER ERROR (red) lights if the parallel printer (if one is connected) has sent an error signal to the HDFS. this usually means that the paper or ribbon has run out, or that the printer is not on-line, depending on the unit in question. In detail. the lamp lights if either the SLCT or FAULT lines (Centronics connector pins 13 or 32 respectively) go to logic low. The exact reasons depend on the printer type.

CONFIGURATION MODE (red) indicates that the system is in the Configuration Mode. where various operations can be carried out, for example backing up to tape. formatting tapes, or setting the Econet station number of the HDFS. This lamp will flash until a computer running the FAST program is communicating with it. This is fully explained in section 3.3.1 below.

ON LINE (green) indicates that the file server is running, and that the system has checked out correctly. If there is still data in the printer buffers, this lamp will flash if the system manager attempts to switch off or enter Configuration Mode. The required operation will occur when the printing has finished, or alternatively the buffer contents may be discarded by moving the front panel keyswitch to another position and back.

3.3 File Server Administration

Since the Hard Disc File Server has no keyboard or screen, all access to it is via the Econet network itself.

There are two methods for the System Manager to gain access to the File Server. The first is to log on to the file server in the same way as any other user, using the *I AM command (q.v. §6), but as a system privileged user. This allows the system manager to perform a range of system privileged operations, which include :

- Reading or writing the Password File using EDITPASS (q.v.)
- Crediting or debiting accounts, using *CREDIT and *DEBIT (q.v.)
- Setting the internal real-time clock using SETTIME (q.v.)
- Setting up the built-in printer server using EDITPRINT (q.v.)

These operations are described in detail in second half of this chapter under the corresponding command specifications.

Two safeguards are provided against any user changing the configuration of the system. The system manager must have system privilege: as set up in the password file. System privileged users are recommended to set passwords to prevent other users trying to log in with their user identifier. Second, the key-switch on the front panel must be turned to the SYST position for all system privileged operations. This means that a key is needed (as well as access to the room in which the HDFS is kept).

The second method of access to the system is for more fundamental operations, including :

- taking backups
- formatting backup tapes
- changing the station number
- changing disc names
- clearing the password file after a disaster

These are all administrative operations of a type which cannot be performed while the File Server is running. In the HDFS this mode of access is called Configuration Mode.

3.3.1 Entering Configuration Mode

To enter configuration mode in the HDFS, turn the front panel key-switch to the SYST position, and press the ENTER CONFIGURATION MODE switch on the rear panel of the HDFS. This is a toggle switch which should be pressed across to the left (as you stand in front of the HDFS unit). The green ON LINE lamp on the front panel will go out: and the red CONFIGURATION MODE lamp will flash. At the BBC Microcomputer fitted with the FAST terminal ROM (see section 3.1.1), type:

***FAST**

FAST will return the prompt :

FAST terminal program 1.03

Station number to attach to :

Type in the File Server station number (usually 254). The red CONFIGURATION MODE lamp on the HDFS will light steadily, and the HDFS will reply with the prompt :

HDFS Main processor
=====
version 0.61

B - boot FS
C - various tests
H - this help msg
N - new disc name
P- clear password file
S - set FS Station number
T - tape routines

SRL's Portable Monitor version 0.8d

A>

3.3.2 Configuration Mode Commands

A range of single letter commands are available as listed below. They should all be followed by <return>.

B boots up the file server program. The unit will read the File Server program from disc, and the green ON LINE lamp on the front panel will eventually light.

C runs various tests, for maintenance only.

H repeats the above menu.

N changes the name of the disc. The system will prompt :

Old disc name is :Main-2
New name (max 10 chars) :

P clears the password file. This will be necessary if it had been accidentally saved without containing a system privileged user with access to account 0. The system will prompt :

About to clear the password file -
Are you sure? :

S sets the station number of the HDFS. The prompt will be :

Current station number is :254
New station number

This command will only have effect when the File Server is booted up again by typing B

T changes to the tape menu, described in detail in Section 3.3.4 below.

There are other commands apart from those listed here. These are supplied for system maintenance. and should not be used by system managers.

3.3.3 Re-setting the HDFS Station Number to 254

If the ENTER CONFIGURATION MODE switch is held across while the system is switched on from STOP, through SECURE to SYST. and then back off again, the station number will be reset to 254 (the normal default file server number). This is made deliberately difficult to do. to avoid accidental re-setting. The re-setting operation should only be required in the unusual circumstance of the system manager having forgotten the station number.

The HDFS station number will be held in memory for as long as the internal battery holds up. about 6 to 12 months. If the HDFS has not been used for this time, it may be found to have station number 254 on re-use -- also the real time clock will have to be reset. as it runs from the same battery. See under SETTIME in section 3.5.

3.3.4 Backing-up the Hard Disc on to Tape

A random access tape drive is provided as part of the HDFS system. Currently there exist commands to back up either the system tracks of the HDFS. or the data (i.e. the files), or both.

To perform a tape backup, enter the configuration mode by turning the front panel key switch to SYST. and pressing the ENTER CONFIGURATION MODE switch on the back panel as described in Section 3.3.1 above. Then type *FAST on the BBC Microcomputer equipped with the FAST EPROM.

If the tape is to be written to (i.e. backup rather than restore): ensure that the write protect screw in the corner of the cartridge is turned away from the SAFE position. Conversely, turn the screw to SAFE on valuable tapes.

Open the tape drive by turning the small lever through 90 degrees clockwise. and pull the sliding tray out to its full extent of about 35 millimetres. Insert a tape cartridge type DC600A into the drive (with the label to the left). push the tray fully home, and turn the lever 90 degrees anticlockwise. The tape will wind through, taking about 3 minutes. The unit will buzz when the tape has finished winding.

On the BBC Microcomputer running *FAST you will receive the prompt **A>**: Type **T** to enter the tape menu:

A>T

Tape options:

N - normal action - ie:

(Format) /Backup /Check tape

C - check tape against disc

F - format tape

V - verify tape

B - backup data/system

R - restore data/system

Ensure that there is a tape in the drive before running any tape command.

The normal command N will first prompt

Do you wish to format the tape ?

Type Y to format the tape. otherwise the system will assume the tape is already formatted. The data and program on the File Server will then be stored on the tape. then checked back against the disc. Each of these operations takes about 10 minutes. making 20 minutes (or half an hour if formatting is requested).

The command F will format a new tape: the system will prompt

Which stream (<CR> does all 6)

Type <return> in response. Formatting takes about 10 minutes.

The B option will perform a backup without formatting the tape. The system will prompt :

Which area:

D - data area only

S - system area only

V - variable

Type D to backup the File Server data, or S to backup the system (i.e. the File

Server program itself)

Re-enter the tape menu with **T<return>**, then type C to check the tape back against the disc.

The data and system are stored on separately reserved areas of the tape, and can be backed up or restored independently. Note however that re-formatting will destroy everything that was previously on a tape.

Typing **V** after the main tape menu will verify that the checksum for the data on the tape is correct, regardless of the data on the tape. The command C is more useful, since it checks back against the original data.

The following error messages can be generated :

During tape reading, the letter **r** means that a checksum error has occurred, and that the system is re-reading that sector. After 4 attempts, **Tape Error 08, DTS=dd,tt,ss** will be printed.

If a tape has not been formatted, or has a substantial drop-out, **Tape Error 10, DTS=dd,tt,ss** will be printed.

During the check tape option, a difference between the tape and disc data will cause **Tape Error FF at block &bbbb**. The system will try to re-write the data on to the tape.

Note: Do NOT open the tape drive while a tape operation is in progress. The green CHANGE TAPE lamp will light when the drive may be opened. Removing the tape during a restore operation will almost certainly leave the disc in a state where it cannot be read back.

3.3.5 Recovering the System or Data from Tape

To recover data from the tape, type R after the tape options prompt. The system will prompt :

Which area:

D - data area only

S - system area only

V - variable

:

as above, and will recover the specified areas from the tape, and write them back to the disc.

(In future versions of the HDFS software, commands will be available for users to back up and recover selected files.)

3.3.6 Maintenance of the Tape Drive

The tape drive needs a small amount of routine maintenance. Every 8 hours of tape motion, the tape head and capstan roller should be cleaned using a 6 inch long lint-free cotton bud (available from RS Components, stock no. 558-802), moistened in Freon TF cleaner (RS Components, stock no. 554-838). Do not use other solvent cleaners.

The positions of the head assembly and capstan roller are shown in the diagram below. Clean the head with a gentle vertical motion of the cotton bud, then repeat with a fresh bud. The capstan should also be cleaned with a vertical motion, then it should be rotated slightly with the cotton bud, and cleaned again -- repeat this process until it is clean all round.

Allow all the solvent cleaner to evaporate before inserting a tape.

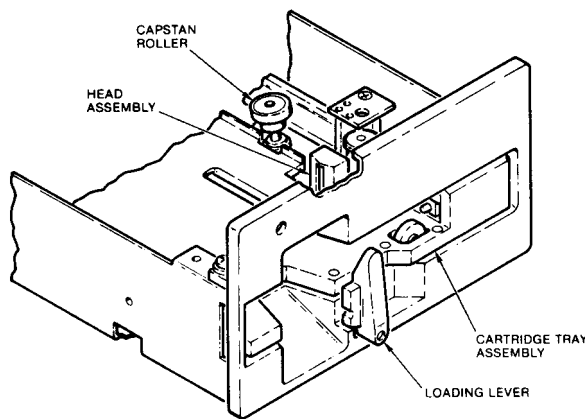


Fig 3. Tape Head and Capstan Roller Location

Suitable tape cartridges for the HDFS tape unit are made by the 3M Company, type DC600A. These are available from SJ Research, or from any 3M tape distributor, for example:

Action Computer Supplies
6 Abercorn Trading Estate
Manor Farm Road
Wembley HA0 1WL
Tel. 01-903 3921

3M's address is

Data Recording Products
3M United Kingdom PLC
3M House
PO Box 1
Bracknell
Berks RG12 1JU

3.4 General Operating Recommendations

3.4.1 Network Security

This section deals with the problems of protecting data on Econet systems generally and on SJ file servers in particular.

There are two separate areas to this question: accidental and deliberate interference. Accidental problems can be further subdivided into 'mistakes' by the users such as saving one program over another, or deleting too many files with an injudicious wildcard; and 'external accidents' such as turning off the file server at the wrong time or a lightning strike. Deliberate interference also divides into two categories. 'snooping' or passive activities to gain access to other peoples data and 'wrecking' or active alteration of the data.

3.4.1.1 Accidents

Mistakes

No-one can guard against mistakes completely but users can easily be protected from many of the more common errors.

The most common ways that beginners lose files are:

1. Saving over a file with a new program of the same name.
2. Saving a 'null' program (in the case of Basic this will be two bytes long) because they have not typed **OLD** after pressing <break>.
3. Using wildcards in a delete statement with more effect than intended.

Encourage users to use names with meaning. Ten characters is enough to chose a sensible name, especially if the use of hyphens and underlines is encouraged.

Suggest that users check their saves with a ***INFO <filename>** after each save, and explain how to check the file length. Alternatively use ***OPT 1.1** to display this information automatically.

Beginners can be given a lot of protection by initially giving their directory a default access which is locked (for example ***DEFACCESS LWR/**). The disadvantage of this is that it is then necessary to type ***ACCESS <filename> -L** before a new version can be saved over the existing one. If you plan to use programs like EDWORD which do not allow * commands then this is a disaster!

Beginners may also appreciate the 'No Short Saves' option which can be set by the system manager using the EDITPASS program. There is also an option to require the use of ***ENABLE** before any wild card delete operation is permitted.

Recovering from Mistakes

Generally speaking: there is no easy way to recover a 'lost' file. In future versions there will be the facility to recover a single file from a tape back-up, but for the present the only way to get back data from a back-up tape is as follows.

1. Take a back-up of the present state of the system. Make sure that it verifies correctly!
2. Restore the old back-up containing the 'lost' file.
3. Load the lost file into a BBC machine, or copy it onto local disc or cassette.
4. Restore the new back-up.
5. Save back the recovered files.

3.4.1.2 External Accidents

The important thing to remember is that computers do crash. Even assuming no software bugs and perfectly working hardware, accidents can and will still happen. Power cuts are a major cause of problems. with lightning a close second.

The first rule to remember is Take frequent back-ups. This applies especially to Econet file servers, where the users themselves have absolutely no control over their file integrity. In many cases a file server may be your only mass storage device. Sooner or later IT WILL FAIL. The only guarantee against disaster is to take a back-up every day.

Reliability of the file server can be improved in other ways too. The ideal environment is a firm shelf in a well ventilated position where people are unable to shake the machine. If possible the file server should be left switched on at all times. Winchester discs are most likely to be damaged during the critical power-on sequence, when the heads are not flying in the normal way but are rubbing on the surface of the disc.

3.4.1.3 Deliberate Interference

Care When Logging on

One of the most common security hazards is using an unprotected machine. Before doing anything sensitive, remember to run *PROT on your computer. In particular. this is worth doing BEFORE logging on as a system privileged user. If you do not do this. your memory can be examined by any other machine with the appropriate software. (Note that this problem is not specific to the SJ file servers. and indeed remains a problem even if there is no file server on the network at all.) Protected status is cleared by <Ctrl-Break>.

Remembering to Log Off after a session:

The file server and your local station both store information about you. Pressing Ctrl-Break or even switching off your local machine will not affect the file server's status: so that only a small amount of trial and error will allow another user to re-enter the system from the same terminal. (You can show this to be so by switching off your own terminal and typing '*USERS' from another machine.) IT IS ESSENTIAL FOR ANY USER WHO IS CONCERNED ABOUT SECURITY TO TYPE '*BYE' BEFORE LEAVING HIS TERMINAL.

It is also good practice to turn off your machine after a session, in case any secret information has been left lying around in RAM (the EDITPASS program is a good example of this).

Passwords

If a user gains access to your files in a 'legal' way, there is very little that you can do about it (but see the next section for one way of limiting the damage.)

The difficulty in ascertaining someone's password by trial and error increases rapidly with the number of letters in the password. Remember too that numbers and other non-alphabetic keys (J, etc) can be used. The password is governed by the same restrictions as for filenames. All passwords should be at least five letters long, preferably more. The maximum length is ten characters. Users should not use passwords which might be guessed easily by others (do not use your wife's name: phone number or car number. nor do we recommend the use of characters from Tolkien or Hitchhiker's !).

Take care that you do not leave your system unprotected whilst you are still learning it : it is very hard to re-establish security once it has been broken. One

enterprising hacker managed to add a command to EDITPASS, to save a copy of the password file into his own directory. By placing a <ctrl-U> character before and a <ctrl-F> character after his code, he ensured that it would not be listed on the screen of a BBC Micro.

Keys and Key Discs

The Hard Disc File Server requires the key switch on its front panel to be turned to the SYST position before any system privileged operations are permitted. Make use of this feature all the time, ensuring that you do not leave the key in the SYST position if you do not require system operations.

'Snooping' and 'Wrecking'

Provided that passwords do not fall into the wrong hands, the opportunities for deliberate meddling can be fairly limited. Obviously it is useful to make all sensitive areas of the disc **Private**. Non-private directories can also be made reasonably secure by setting the access to **WR/**. Note that if the Default access includes public access, there will always be a period during the creation of a new file when other people can read it, even if it is then rapidly set to WR/.

Care when not using the network

Remember that the network is still active, even if you are not using it. For example if you are editing files to a local disc, your screen can still be viewed ! This could have serious consequences if you were writing exam papers for example. Use the program *PROT (preferably in your boot file where it will be run every time you log on) to avoid all but the most dedicated hacker. The 100% safeway to avoid hacking is to unplug your system from the network when doing work of a particularly sensitive nature.

3.4.1.4 Special Techniques

Limiting the Opportunity for Damage

Editing the password file may only take place when the file server key-switch is in the system position. This editing allows a system privileged key-holder to remove everyone's access to a particular account (including his own access) and then 'lock' the password file. In this state, no-one may access private files that have been saved in that account, not even the key-holder. The password file will have to be re-edited to enable access again. This technique may have some use for files which are frequently read but only rarely changed (e.g. libraries or time-table information), which may be made WR/R and saved into an account which is not normally accessible. Take care not to remove your own access (as system manager) to account 0, otherwise you will not be able to edit the password file.

Restricting read access to files and restricting write access more:

General Principle: Allocate two accounts, one for read access only and one for write access.

Create a private directory in the Write account, with auxiliary access in the Read account and Defaccess WR/R. Only those users with access to one or other (or both) of the two accounts will be able to see or enter this directory.

When a new file is created within this directory, set the auxiliary account to the write account, thereby removing from the read account the owner access to the files.

giving them public. 'read-only' access rights.

Example: Pupil Reports are required to be available for inspection by any member of staff, but only the master in charge should be able to alter them. The master in charge should have accounts 2 and 3. the other staff account 3. and the rest of the users access to neither account.

Method:

1. The creator selects the place where he wishes the new directory to appear.
2. *CDIR reports
3. *ACCOUNT reports 02 (03)
4. *DIR reports
5. *DEFACCESS WR/R

The directory is now ready for use. When saving a new file. type

6. *SAVE <file name> 0+0
7. *ACCOUNT <file name> (2)
8. *ACCESS <file name> WR/R

then save the file you want as <file name>. This avoids a malicious user reading the file between it being saved and its access changed.

3.5 Command Specifications.

This section contains a detailed description of the commands and programs intended for the use of the system manager for the day to day running of the File Server system.

These commands are not destructive in the wrong hands : most of them require system privilege to have any effect (although the programs can usually be started by anyone). In addition, the front panel key switch must be in the SYST position .

The commands *BANNER. *ENDTEXT. *PRE and *POST have now been withdrawn, and the banner for a printer is now specified in a file, whose name is given to the program EDITPRINT (q.v. this Section). The banner file specification is given in Section 3.6.

The commands and programs documented here are :

*CHECK	Machine code program
*CREDIT	File Server command
*DEBIT	File Server command
EDITPASS	BASIC program
EDITPRINT	BASIC program
*FAST	Machine code program
*FLUSH	File Server command
FORCER	BASIC program
*LOGOFF	File Server command
MONSWITCH	BASIC program
*NETMON	Machine code program
SETTIME	BASIC program
SIZER	BASIC program

***CHECK**

Command Type : Machine code program

Syntax : ***CHECK**

Action with Wild Cards in the File Name : Not applicable.

Description :

This program will check the issued library utility programs using a 'checksum' algorithm. and will compare them with internally held values. It contains its own list of files to check. and it will list as it runs. and print a warning message if any of them has been tampered with.

The files currently loaded and checked are :

BUILD. CLOSE. COPIER. CV. DISCS. DUMP. EDITPASS. EDITPRINT. ERAQ. FAST. FREE. FSLIST. GO, LOGON. MONSWITCH. MULTICOPY. NETMON, NOTIFY, PS. PRINT. PROT. PTIME. PDATE. PDATE2, PUTGET. PUTGET2, PRL.PUTGET. SETTIME. SIZER. STATEMENT. STATIONS, TYPE. UNPROT. USERS. VERS.

Each file will be listed, with no further comment if its checksum was correct. If it was not correct, the message **This version old or corrupted** will be displayed. If the file was not found. **File not found** will appear.

If the currently selected directory is likely to contain any files of the same name as these utilities. then it is wise to type ***DIR \$.LIBRARY** before running CHECK.

CHECK is encrypted to make it difficult for casual hackers to modify it. and it performs a checksum upon itself when run. but system managers are recommended to keep it on DFS floppy disc, locked away. and perhaps not leave it anywhere in the File Server.

CHECK will be re-issued with any new version of the library.

Examples

***CHECK**

BUILD

CLOSE

COPIER This version old or corrupted CV

File not found

Likely Errors :

If the program's own checksum is incorrect, then the message **This program has been meddled with !** will be displayed.

Compatibility Notes :

Not supported by Acorn systems. as it is specific to SJ utility programs.

***CREDIT**

Command Type : File Server command.

Syntax :

***CREDIT <account number> <amount>**

Action with Wild Cards : Wild cards prohibited.

Description :

This command adds the specified <amount> in units of 1 kilobyte (K) to the balance of account <account number>. Account numbers range between 0 and FF and are in hexadecimal, but the amount of credit is in decimal.

The maximum balance possible is 65535K, and the system manager may effectively turn off the space accounting system by setting accounts this much credit. 65536K is the largest single disc that can be fitted to a system.

System privilege is required to execute this command.

The accounts system gives control of disc space filled by users' files. This command increases the allocation of space available for files with the specified account number.

Note that the current credit balance of an account represents only the free space available for files, and does not include files already existing with that account. It is therefore wise for the system manager to keep a notebook containing details of users, their accounts and any subsequent credit or debit given to them. A utility program SIZER is provided to find the total size of files in a directory tree, and this could be used to find the total space taken up by a user's files at a later date.

The transient program *STATEMENT will give a printout of all the accounts to which a user has access, with the associated credit balance for each. The system manager will normally have access to all accounts, and so will get a long list of all accounts from 0 to FF for each disc in the system. It may be wise to send this output to the printer if all the information is required.

Examples :

***CREDIT 89 512**

allocates a further 512K to account number 89

Likely Errors :

Error 04 (4) Bad number -- if there is not a number specified in both fields or the number is in the wrong base (hex for the account number, decimal for the amount).

Error 05 (5) Key locked -- if the key switch on the front panel of the HDFS is not in the SYST position

Error BA (186) Insufficient privilege -- if the user does not have system privilege.

Compatibility Notes :

Not supported by Acorn systems

***DEBIT**

Command Type : File Server command.

Syntax :

***DEBIT <account number> <amount>**

Action with Wild Cards : Wild cards prohibited.

Description :

This command subtracts the specified <amount> in units of 1 kilobyte (K) from the balance of account <account number>. Account numbers range between 0 and FF (hexadecimal)

The maximum balance possible is 65535K, so the system manager may wipe out the balance of an account ***DEBIT <account number> 65535**.

System privilege is required to execute this command.

The accounts system allows control of disc space filled by users' files. This command decreases the allocation of space available for files with the specified account number.

Note that the current credit balance of an account represents only the free space available for files, and does not include files already existing with that account. It is therefore wise for the system manager to keep a notebook containing details of users, their accounts and any subsequent credit or debit given to them. A utility program SIZER is provided to find the total size of files in a directory tree, and this could be used to find the total space taken up by a user's files at a later date.

The transient program *STATEMENT will give a printout of all the accounts to which a user has access, with the associated credit balance for each. The system manager will normally have access to all accounts, and so will get a long list of all accounts from 0 to FF for each disc in the system. It may be wise to send this output to the printer if all the information is required.

Examples :

***DEBIT 89 512**

removes 512K from account number 89

Likely Errors :

Error 04 (4) Bad number -- if there is not a number specified in both fields or the number is in the wrong base (hex for the account number, decimal for the amount).

Error 05 (5) Key locked -- if the key switch on the front panel of the HDFS is not in the SYST position

Error BA (186) Insufficient privilege -- if the user does not have system privilege.

Compatibility Notes :

Not supported by Acorn systems

EDITPASS

Command Type : BASIC program.

Syntax :

CHAIN "EDITPASS"

Action with Wild Cards : Not applicable.

Description :

The system maintains a special file %PASSWORDS, which does not exist in any particular directory. and is only 'visible' to system privileged users. The password file cannot be deleted with *DELETE. but will have to be cleared in Configuration Mode, although it is possible to save over the top of it. The name must be quoted in full in upper case, i.e. no wildcards. EDITPASS is provided in the library for editing the contents of %PASSWORDS.

EDITPASS is a screen editor for editing password files, or for creating new ones. The present version can handle about 200 users, and requires a BBC computer with 32K RAM (i.e. Model B or expanded Model A).

[There is a version of this program, EDITPASS!, which uses more readable identifiers, if the system manager wants to see how the program works or to make his own version. EDITPASS itself has been condensed, so as to leave maximum store for data].

Although anyone can run EDITPASS (assuming that the system manager has set public read access to this file), it is necessary to have system privilege and access to account 0. and the HDFS front-panel keyswitch must be in the "SYST" position to either read or save the password file on a disc.

EDITPASS always works with the currently selected disc drive . to edit the password file on another drive, use *DIR .<disc name> to select this new disc.

Before running this program, run the utility ***PROT**, to prevent another user from using *VIEW (or otherwise) to read your computer's memory.

When the program is run, the program will prompt

Password file Editor 20/1/85

Maximum number of users=257

Options:

E - edit PW file from disc

N - create new PW file

O - edit file from current RAM

The normal option is **E**, unless it has been necessary to delete the password file for some reason. The **O** option is useful if this program has stopped, either with an error, or as a result of pressing the <escape> or **Q** key.

The program will then display all the users, with their boot options and system privilege (if any). The display will be similar to :

A(dd) Delete) S(ave) X(pand) Q(uit) *

User id	Boot option	Privilege
ARG	0 Off	
BASHER	0 Off	System
BJ	0 Off	
BOOT	3 Exec	
CLAIRE	0 Off	
DEFAULT	2 Run	*DEFAULT*
JEF	0 Off	
JOHN	3 Exec	
KIM	2 Run	

The 'up' and 'down' cursor keys scroll the display, allowing the total user list to be available. To add new users, press the A key : the program will prompt repeatedly for names until the list is terminated with <return> on its own.

D key will delete the user on the current line.

* key allows normal * commands to be run from within the program. To return to the list of users, type <return> on its own on a line.

Q key stops the program without saving the result back to disc. The <escape> key, pressed at any stage, has the same effect.

S key saves the password file to disc, and stops. A check will be made that there is at least one system privileged user, and that a user exists with access to account 0 (these are both vital to the running of the system), and the program will print a warning if one of these requirements is not met. Note that you may however want to keep all your system users on one (or a few) discs for security, in which case it would be legitimate for there to be no system user, or user with access to account 0. on other discs. The name of the default user if any (see below) will be displayed, and then the prompt **Save (YIN)** : Typing **N** will return to the list of users.

After typing Q or S, at the end of the system manager's session, type ***BYE** then TURN OFF THE BBC MICROCOMPUTER at which you were logged on. The password file will remain in the computer unless it has been overwritten, and another user could easily read it from there, and gain unauthorised access to the system.

X key enters the expanded mode for the user at the current cursor position. The display will change to, for example :

```
A(ccounts) S(ystem priv) D(efault user)
P(assword) U(RD) L(IB) 0..3 Boot option
8-Saves 9-PW lock E(nable)
```

```
ARG                      0 Off
PW : None
URD: None                 Short SAVES allowed
LIB: None                 Permanent *ENABLE
```

```
Accounts :             0->15 25 60->6F F0->FF
```

Expanded mode allows the details of each user to be edited. New users are initially created with no password set, boot option 0 (see under *OPT4, §6) and access to no accounts at all.

0 1 2 or 3 will set the boot option to that value. The boot option may also be set by the users themselves, unless the **PW lock** option has been set (see below):

- 8** will turn on and off an option to prevent users from saving a file shorter than 16 bytes in length with the SAVE or *SAVE command. This option helps to avoid the problem where BASIC (for example) will save a null file if an attempt is made to save after pressing BREAK without typing "OLD"
- 9** will turn on and off an option to prevent users from changing their password and boot option. It could be useful to set this option for the user BOOT, to prevent unauthorised users from changing these.
- S** toggles system privilege off and on for that user. Note that there must be at least one system privileged user on the system, or it will not be possible to change the password file hereafter.
- D** sets this user as the default user. There can be only one default user, so this command will change the default to this user. Keying D again will remove the default setting, so that there is no default user. Users logging-on to the File Server with unrecognised user identifiers will be logged on as the default user the system manager should set up a boot file to re-direct them, if necessary. If there is no default user, the error **User not known** will be displayed.
- E** toggles an option requiring the user to type *ENABLE before attempting *DELETE with a wild-card specifier, as a safety measure.
- P** prompts for a password. Users can also set up their own passwords with the *PASS command unless the **PW lock** option has been set.
- A** prompts for account numbers. In response it is possible to specify a single account, several accounts separated by commas or spaces, or a range of accounts: for example 1-55 specifies all accounts from 1 to 55. If the line starts with a minus sign, the specified account(s) are removed from the user's list, otherwise they are added. Note that account 0 is a 'system' account as the root directory on each disc always has account 0 : hence account 0 should normally only be allocated to the system manager.

To return to the list of users, press <return>.

(The keys **U** and **L** in expanded mode refer to the future facility to select default libraries and user root directories, and are currently not supported).

Likely Errors :

Error 05 (5) Key locked -- if the key switch on the front panel of the HDFS is not in the SYST position
 Error BA (186) Insufficient privilege -- if the user does not have system privilege.

Compatibility Notes :

Not supported by Acorn systems.

EDITPRINT

Command Type : BASIC program

Syntax :

CHAIN "EDITPRINT"

Action with Wild Cards : Not applicable.

Description :

The **EDITPRINT** program allows the system manager to set up details for the two printers which can be connected to the File Server.

Although anyone can run EDITPRINT (assuming that the system manager has set public read access to this file), it is necessary to have system privilege , and the HDFS front-panel keyswitch must be in the "SYST" position to either read or save the printer information on a disc.

Two printers can be connected to the File Server -- section 3.1.2 gives connection details and other instructions for the installation. This program is used to give the printers names by which users can refer to them, and to set up other rules for access to the printer(s).

When the program is run, then initial menu will read :

Edit printer details	- 1
Change priority order	- 2
Select system message channel	- 3
Reset print server & exit	- R ETURN

Enter choice :

Enter Printer Details

Choosing 1 from the main menu will give the further prompt :

Which printer (1=parallel, 2=serial)1

then the options menu will be displayed, for example :

Printer 1

1 - Name : PARALL
2 - Printer exists : Yes
3 - Anonymous users allowed : Yes
4 - Account ownership required : No

6 - Banner file name :

Change which item (1-7, RETURN to save changes) :

The printer can be given a name of up to six characters -- more characters may be entered, but only the first six will be recorded or subsequently checked. The names PRINT and AUTO are reserved (see section 6.5 on printing) and may not be given to a printer.

Option 2 tells the printer server whether this printer exists -- if it is set to not exist, then it will not be possible to send output to it.

Option 3 sets whether anonymous users -- i.e. users who are not logged on to the File Server -- are allowed to use this printer. Note that such users may not select printers by name, but can only access the automatic selection as set up below (see

also section 6.5)

Option 4 can be used to permit only users who have access to a particular account to use this printer. If this option is set to Yes, then option 5 will prompt for the account number. This facility is useful for restricting the access to particular printer devices, for example a plotter or a daisy-wheel printer.

Option 6 tells the system the file name containing the banner and end-text information for the printer server. The format required for this file is described in Section 3.6. The name of the file specified here may not exceed 23 characters in length. Unless a disc name and/or root directory are explicitly specified, the root directory \$ on Drive 0 will be assumed. The two printers may have the same file as banner file, or different files. If no file is specified, then no banner or end-text string will be printed.

The system must have read access to the banner file (i.e. there must be letter R in either the owner or public access), otherwise no banner or end-text will be printed. This is also the case if the file is not found.

Change Priority Order

Number 2 from the main menu allows the system manager to select which printer(s) are selected by a user typing ***PS AUTO** (or not typing ***PS <name>** at all), and the order in which they are allocated. The system manager would normally set up the fastest text printer available for first choice, and perhaps a slower one for second choice. Printer number 0 means that no printer is selected, and would be used to prevent output going by default to a plotter, for example. The menu is :

This option allows the system manager to select the order in which printers are selected for users who have not requested a particular printer.

Current order :
1st choice is printer 1
2nd choice is printer 0

New 1st choice
(number of printer, or 0 if none) : 1
New 2nd choice
(number of printer, or 0 if none) : 2

Select System Message Channel

Number 3 from the main menu allows the manager to select the printer at which system messages are sent. These messages can be sent to the same printer as users' output, since they will be separated by a page throw and a header from other output.

On the other hand, it is also permitted to specify here a printer which was set as 'non-existent' in Option 1 above. This will prevent users ever being able to select this printer, and effectively reserves it for messages only. This course is recommended if full messages (i.e. full log of all File Server operations) have been turned on. The program MONSWITCH (q.v. this Section) controls the scope of messages sent to the printer.

If an error occurs at power-up, any message from the system will be sent to the parallel printer always. This is done because the printer choices are stored on disc, and an error of this type will normally occur because it was not possible to read the disc properly.

The menu for setting the system messages destination is:

**System messages currently go to
printer 1**

New destination : 2

Although the system message channel can be set to 0, system messages will still appear at printer 1. This is because the system message can be important in the case of system failure.

Likely Errors :

Error 05 (5) Key locked -- if the key switch on the front panel of the HDFS is not in the SYST position

Error BA (186) Insufficient privilege -- if the user does not have system privilege.

Compatibility Notes :

Not supported by Acorn systems.

***FAST**

Command Type : Machine code utility program, supplied in the library and in EPROM

Syntax : ***FAST**

Action with Wild Cards : Not applicable.

Description :

This program turns the BBC Microcomputer into a terminal to any suitably equipped host computer : communication is through the Econet network.

The most common use of *FAST is to communicate with a SJ Research File Server system, when the latter is in Configuration Mode.

The program will prompt :

Station number to attach to :

Type in the station number of the File Server or other host computer (usually 254).

The version of *FAST in the library runs in normal (RAM) memory, and can be used by anyone who has read access to the file. The version in EPROM will be fitted to one of the BBC Microcomputers on the network, as part of the installation process. Either version can be used, but the library (RAM) version needs to be started before the File Server is taken off line, and put into Configuration Mode. In this case, type *FAST, and the station number, but press <return> only after the File Server is in Configuration Mode.

The EPROM version will re-start if <Break> is pressed : it will be necessary to press <ctrl-Break>, or to type ***BASIC** after the **Station number** prompt, to exit from the program.

The RAM version will stop if <Break> is pressed. It may be re-started by typing **CALL &2000**.

In either version, * commands may be entered at any time, preceding them by <Shift-fl>, or by typing *<command> after the **Station number** prompt:

Examples :

***FAST**

Station number to attach to : 254

followed by output (if any) from station 254:

Likely Errors :

Normal BBC Microcomputer errors will be preceded by **OS Error**.

No response is usually caused by the File Server not being in Configuration Mode, or by being attached to the wrong station.

Compatibility Notes :

Not supported by Acorn systems.

***FLUSH**

Command Type : File Server command, controlling the built-in Printer Server

Syntax :

***FLUSH**

Action with Wild Cards : Not applicable.

Description :

This command causes the contents of both printer buffers to be thrown away.

It will be found useful if a user's program has generated large quantities of spurious output.

System privilege is required to use this command.

Note that printers themselves often have an internal buffer, which means that they could carry on printing for some pages after a *FLUSH command. To clear the internal buffer, it will be necessary to turn the printer off and on.

Likely Errors :

Error 05 (5) Key locked -- if the key switch on the front panel of the HDFS is not in the SYST position

Error BA (186) Insufficient privilege -- if the user does not have system privilege.

Compatibility Notes :

Not supported by Acorn systems.

FORCER

Command Type : BASIC program

Syntax :

CHAIN "FORCER"

Action with Wild Cards in the File Name : Not applicable.

Description :

This program sends a string to the keyboard input of one or more stations -- i.e. it will force them to execute any command sent. It could also be used to send a message (preceded by *| which causes the O.S. to ignore the rest of the line).

It is possible to send the command to all stations of a group. Load FORCER (it is in the release library), and re-type line 50 to contain all the stations which are to be included in the group :

50 DATA <station number>,<station number>, ... END

When the program is run, it will prompt

Station (<RETURN> for all) :

then,

Command :

If <return> is typed after the first prompt, then the command will be sent in turn to all the stations in the DATA statement:

Examples

CHAIN "FORCER"

Station (<RETURN> for all) :

Command : LOAD "EXAMPLE!"

Sending to station 1

Sending to station 3

It is wise to hide away the *PROT utility that sets the protection byte to stop direct operations, otherwise users can prevent FORCER working on them.

Likely Errors :

The program uses the same Osword call as *NOTIFY, and can give rise to the same errors, e.g. **Not listening** if one of the stations is absent, switched off or protected.

Compatibility Notes : Supported
by Acorn systems.

***LOGOFF**

Command Type : File Server command

Syntax :

***LOGOFF I *LOGOFF <user identifier>{,<user identifier>}**

Action with Wild Cards : Will apply the wild-card to user names, to log off all occurrences of all matching user identifiers

Description :

Used without a parameter, this command has exactly the same effect as ***BYE** (see §6.6)

When followed by a user identifier, or a list of user identifiers separated by commas, all occurrences of the users (except for the station originating this command, if that user is included in the list) in the system will be logged off.

Wild cards are permitted, so that

***LOGOFF #IM**

will log off all occurrences of users JIM, KIM and TIM etc. The command ***LOGOFF *** will log off everyone except the person originating the command.

Anyone may use ***LOGOFF** without a parameter, otherwise system privilege is required.

Likely Errors :

Error 05 (5) Key locked -- if the key switch on the front panel of the HDFS is not in the SYST position

Error BA (186) Insufficient privilege -- if the user does not have system privilege.

Compatibility Notes :

Not supported by Acorn systems.

MONSWITCH

Command Type : BASIC program

Syntax :

CHAIN "MONSWITCH"

Action with Wild Cards : Not applicable.

Description :

This program turns the File Server messages output on or off. The messages are sent to a printer set up by the program EDITPRINT (q.v.).

Although anyone can run MONSWITCH (assuming that the system manager has set public read access to this file), it is necessary to have system privilege, and the HDFS front-panel keyswitch must be in the "SYST" position to change the printer information.

The effect of MONSWITCH lasts until it is run again to change the monitor output, or until the File Server is switched off -- in this case the Monitor and Debug outputs will be off when the File Server is restarted.

This program will produce the output :

File Server Monitor Output Switch

This program controls the scope of messages output from the HDFS printer port.

Normally both Monitor and Debug will be off. Extra output slows down the System, especially the Debug output.

**Do you want Monitor output? (Y/N):
Do you want Debug output? (Y/N):
OK**

The Monitor output will produce a line of print for every filing operation using **LOAD SAVE RUN** and File Server commands. These lines are of the form

Station 006 (FRED): SAVE PROG1

This output is most useful if a log needs to be kept of File Server activity, perhaps if some users were persistently abusing the system.

The Debug output produces full diagnostic information for the File Server, with full details of disc read and write operations and other associated information. It is unlikely to be of use to the system manager, and will slow down the File Server significantly.

If either the monitor or debug options are on, it is wise to reserve the printer for this output, by setting it to 'non-existent' with EDITPRINT. This will prevent the printer server from using this printer, which would cause users' printer output to appear on pages between those containing system messages.

Likely Errors :

Error 05 (5) Key locked -- if the key switch on the front panel of the HDFS is not in the SYST position

Error BA (186) Insufficient privilege -- if the user does not have system privilege.

Compatibility Notes :

Not supported by Acorn systems.

***NETMON**

Command Type : Machine code program

Syntax : ***NETMON**

Action with Wild Cards : Not applicable.

Description :

This program monitors the network continuously, and displays on the screen each data packet that is sent. It is not necessary to have system privilege to use NETMON, but system managers may wish not to give public access to it. It is simultaneously an invaluable tool for programmers writing Econet systems, and a useful 'hacking' aid.

See Chapter 7 for a description of Econet protocols, and further information about this program.

The display can be stopped temporarily by holding down the space bar.

To exit from NETMON, it is wise to turn the computer off and on again, then log on again.

Note that NETMON effectively disconnects the computer from the network, so that direct operations from elsewhere will not work on this computer.

Likely Errors :

None specific to this program.

Compatibility Notes :

Supported by Acorn systems

SETTIME

Command Type : BASIC program.

Syntax :

CHAIN "SETTIME"

Action with Wild Cards : Not applicable.

Description :

This program sets the internal real-time clock in the File Server .

The program will prompt for the current date and time, and will then ask the user to press the space bar to set the clock to the time entered, allowing accurate synchronisation with the speaking clock or Greenwich pips.

Example :

The program will prompt as follows :

**This program sets the File Server clock
It may only be used by Privileged Users**

Use *TIME to see the present setting

Year : 1984

Month (1=Jan,12=Dec) :9

Day (1-30) :9

Hours (0-23) :9

Minutes (0-59) :18

Ready to set time to:

9/9/84 9:18

Press SPACE to set time, ESCAPE to abort

To set the time again with the same value, type GOTO 370 after running the program (but note that this line number may change in different versions of SETTIME).

Likely Errors :

Error 05 (5) Key locked -- if the key switch on the front panel of the HDFS is not in the SYST position

Error 90 (144) Bad time -- caused by an attempt to set the clock to an invalid time or date.

Error BA (186) Insufficient privilege -- if the user does not have system privilege.

Compatibility Notes :

Supported by Acorn Level 3 systems.

SIZER

Command Type : BASIC program.

Syntax :

CHAIN "SIZER"

Action with Wild Cards : Not applicable.

Description :

This BASIC program searches through a directory tree, finding every file and sub-directory. It will print out every file with its size in units of 1 kilobyte, and then give a grand total.

Anyone can use SIZER, but access to all accounts is recommended if the root directory \$ is specified as the directory to search.

The size of a file is not necessarily the same as its extent on SJ Research File Servers. Since files are allocated space in 1k blocks, the size found by SIZER (= the number of blocks allocated x 1 K) will be greater than the extent of the file.

In some cases though the size of the file may be less than the extent. This is because a file may be opened, some information written, and then the pointer moved by a substantial amount before further writing. Such sparse files will be allocated space in 1K blocks only as it is required, and the size of the file as found by SIZER will be only that of the blocks actually allocated.

Example :

Directory to find size of?\$	
Directory \$	4k
Directory ACCOUNTS	1k
Ashton	1k
Total	568k

Likely Errors :

Error 05 (5) Key locked -- if the key switch on the front panel of the HDFS is not in the SYST position (see Error BA below)

Error BA (186) Insufficient privilege -- If the root directory \$ is specified, then SIZER will include the password file in its search. If the user does not have system privilege, this error will be caused.

Error BD (189) Insufficient access -- if there are files or sub-directories without public access R, and the user of SIZER does not have access to the account number of these.

Error D6 (214) xxxx not found -- if there are files or sub-directories with access letter P. and the user of SIZER does not have access to the account number of these.

Compatibility Notes :

Not supported by Acorn systems.

3.6 Specification for the File containing Printer Server Banner and Endtext

The printer server reads its information for the banner that is printed at the head of each user's output, and the text string that is printed at the end, from a banner file containing information as set out below:

The name of the file to be used is set up with the EDITPRINT program (q.v. §3.5). Option 6 prompts for this name. The maximum length of file name that may be specified to EDITPRINT is 23 characters in total, and the default directory assumed is the root (\$) on disc drive 0.

A directory called BANNERS is supplied in the root, containing (initially) two sample banner files, called SIMPLE and FANCY. The banner for each printer is initially set to no file, and the system manager will have to select one of these two files, or create one of his own, in order to get printer banners at all.

A different banner file may of course be specified for each printer (or the same file may be specified for both). If no banner file has been specified, or the file specified is not found, or the system has not got read access to it (this means that the file must have letter R owner access), then the text will be printed without a banner or endtext : no error message will be produced.

It is possible to allow selected users to change the banner for their own purposes, by giving them access to the account of the banner file (preferably having set this account to a unique value, so that the users in question cannot change any other files). It is recommended that the user uses the **RENAME** command, to change the name of the banner file to something else, and again to rename the desired file as the banner file: this avoids the chance of accidentally deleting the main banner file. The users who are allowed to change the banner file will have to be responsible themselves for changing it back after they have finished.

The selected banner file must contain the end-of-text string first, followed by the identifier **<BANNER>** (enclosed in angle brackets as shown), followed by the banner string: **** is synonymous with **<BANNER>**.

The two strings may contain special identifiers from the list below: Note that all the identifiers are enclosed in angle brackets <>. A carriage return character in the text will be entered as a carriage return in the banner -- note that if your printer does not do an automatic line-feed after each carriage return, then a line feed character (|J) should be put each the carriage return character (unless you intend to over-print lines). The character | has a special effect, namely

|@ |A |B etc through to | enters the 'control' character Ctrl-@, Ctrl-A etc. Please consult the manual for your printer to find the correct 'control' characters for any special functions. Many printers require the control character <escape> -- this can be entered by typing |||C.

|? inserts character &7F (delete)

! inserts the ASCII value of the next character, but with &80 added (i.e. 'top bit set' characters). For example, |1A will insert character &C1 (decimal 193)

| followed by any other character inserts that character.
in particular, |< is the only safe way to insert the character <

3.6.1 Special Identifiers

These special identifiers must be enclosed in angle brackets as shown. Upper and lower case characters are treated as equivalent. An unknown word after an open angle bracket will be printed.

If an angle bracket is required in the text string, use |< as described above.

3.6.1.1 Time and Date

These three identifiers do not cause anything to be printed directly, but they select which of three possible times are printed when the identifiers <HOURS>, <MINUTES> etc. are used.

<NOW> Select the current time of day, at the moment when printing is actually taking place. Note that <NOW> will give a different time in the endtext, from that in the banner, owing to the time taken to do the actual printing.

<START> Select the time of day at which the printing job was initiated, from the user's computer. Note that, especially in installations that implement printer spooling, this (and <END> below) may be substantially earlier than the time given by <NOW>

<END> Select the time of day at which the user finished sending characters to the printer.

The following identifiers cause part or all of the time and date as selected above, to be entered into the banner or end-text string. The default time selection is <STAR T>.

<HOURS> gives a two digit hour in the 24 hour clock, with leading zero printed. For example 6 pm will be printed as **18**

<H> is a synonym for <HOURS>

<MINUTES> gives the minutes past the hour as two digits between 00 and 59, with leading zero printed.

<M> is a synonym for <MINUTES>

<SECONDS> gives the seconds past the minute as two digits between 00 and 59, with leading zero printed.

<S> is a synonym for <SECONDS>

<12HOURS> gives the hour in the 12 hour clock, with leading zero replaced by a space.

<AM> gives **am** or **pm** as appropriate. Note that noon is deemed to be **pm**

<DATE> The day of the month, as two digits, with leading zero replaced by a space.

<ST> gives the correct suffix to the day of the month. For example, on the first of the month, the string inserted by <ST> will be **st**, on the second, **nd**, on the fourth **th** and so on.

<MONTHNAME> gives the full name of the month, e.g. **January**. No leading spaces are printed.

<MTH> gives a three letter abbreviation of the name of the month, beginning with a lower case letter. January will be printed as **jan**

<MONTH> gives the number of the month as two digits with leading zero printed. January will be printed as **01**, and December as **12**

<YEAR> gives the last 2 digits of the year. 1985 will be printed as **85**

3.6.1.2 User, Station etc.

<USERNAME> gives the user identifier logged on at the station that originated the print job. No leading spaces are printed.

<STATION> gives the number of the station that originated the print job. The station number is printed with leading zeroes and with the network number (if the station was on a different network), but no leading spaces are printed. For example, station 2 on the same network will be printed as 002, but station 43 on network 7 is printed as 007.043

<BANNER> The delimiter between the end-of-text string (which should appear in the file first) and the banner string proper. See text above for full details.

 is a synonym for <BANNER>

<MARK> gives a reference point for <TAB> (see below).

<TAB nnn> pads out to a position nnn spaces from the last <MARK> identifier. There must be one space (only) between the word **TAB** and the number. If no <MARK> had been given, this command pads out to a position nnn spaces from the beginning of the text. Note that a carriage return does not reset the value of <MARK>, and that only the least significant byte of nnn is read. <TAB 0> is illegal: the instruction will be ignored and the word <TAB 0> will be printed. If the number after TAB is less than the current character position, then the tab will move to the position 256+nnn.

<RESET> [Not yet implemented. Will eventually be available to send a RESET pulse to the printer.]

<PAUSE nnn> [Not yet implemented. Some printers require a pause after a reset pulse, during which no characters are sent.]

3.6.2 Creating the Banner File

To create suitable files, there are 3 possible methods :

1. Use the *BUILD command (documented in Section 6.6). This is the simplest method, but does not allow a single line with more than 25.5 characters. If this is going to be a problem, then use method 2 or 3.
2. Write a short BASIC program that calls *SPOOL <file name>, then outputs the required text using PRINT. then closes the file using *SPOOL on its own. For example :

```
10 *SPOOL BannerFile
20 PRINT "|L<BANNER>|N<USERNAME>          <USERNAME>          <USERNAME>
   <USERNAME>          <USERNAME> |M|J";
30 PRINT "<START><H>:<M>:<S> on the <DATE><ST> of <MONTHNAME>
   19<YEAR>|T|M|J";
40 *SPOOL
```

This program will generate a banner file containing the text in the example below:

3. Use a word processor, for example Wordwise. Do not use a WYSIWYG (What You See Is What You Get) word processor like Acornsoft View, because this generates invisible 'control' characters in the text, which will have unexpected effects on the printer.

We recommend that you do not use the automatic line-feed option available on most printers. If the printer always does a line feed when it receives a carriage return character, then users do not have the option of over-printing lines if they wish. In addition, it will not be possible to print files generated by *SPOOL without double-spacing. (See description of *PRINTOUT command in §6.6)

3.6.3 Example

A banner file containing this text will cause the user identifier to be printed in bold 5 times at the head of the print output, followed by the date and time at which the printing was started by the user. At the end of the user's output, a single page throw (<ctrl-L>) will appear.

```
|L<BANNER>|N<USERNAME> <USERNAME> <USERNAME> <USERNAME> <USERNAME>  
|M|J<START><H>:<M>:<S> on the <DATE><ST> of <MONTHNAME> 19<YEAR>|T|M|J
```

The control codes shown are suitable for an Epson printer: |N means 'start double sized text' and |T means 'start normal text'. If your printer has the auto-line-feed option turned on, omit the |J (line-feed) characters from the file.

4 Econet Error Messages

This section describes messages reported by the BBC Microcomputer in response to network-related errors, with their probable causes. Most of the error messages are generated by the File Server, but a few (No Clock, Not listening, No reply, Line jammed and Net Error) can be caused by network hardware or local software problems. In this case, see also Section 4.4, which gives a fault-finding guide.

To find out the number of any error as given by the BBC micro, type **PRINT ~ERR**. (Remember that "Escape" is usually an error too). **REPORT** will print the text of the last error message.

Although the error numbers given below are those returned by the file server, they are not necessarily the numbers finally seen by the user's program (the variable ERR in Basic, for example). Any error number returned from the file server which is less than &A8 will be converted to &A8 by the local Net Filing System, although calling OSWORD with A=&13 and reason code=10 will give the actual value (See Acorn Econet User Guide Section 11.8.1 or Econet Advanced User Guide page 62). Locally generated errors (see the list above) will not be subject to this translation.

Some of the less common errors do not produce an explanatory message but instead produce 'FS unusual error xx', where xx is the hexadecimal error number.

SJ Research file servers use the same error numbers as Acorn for corresponding errors, but will also use additional ones in connection with the further features offered.

In the list below, the error number is given in hexadecimal as xx, with the decimal equivalent in brackets.

4.1 Network Related Error Messages generated locally

Error A0 (Decimal 160) - Line jammed

The **Line jammed** or **Net error** messages will occur for one of two reasons. First, there could be a fault in the network terminator (check that no-one has unplugged it!) or in the line itself. This is unlikely if the network has worked in the past, unless there has been some mechanical damage to the cable. Second, there may be a fault in one of the computers on the network. Unplug computers one at a time until it is possible to communicate again. If the problem occurs on only one computer, suspect that particular machine, which could have a fault in its network interface. See Section 4.4 for further information

Error A1 (Decimal 161) - Net error

Net error tends to be caused by similar things to **Line Jammed**, see the explanation of **Line jammed** (Error A0) above. In addition, **Net error** can be caused by sending a data packet which is too large for the system. An example is the use of *NOTIFY with a line of more than 80 characters.

Error A2 (Decimal 162) - Not listening

The "Not listening" message happens when the station to which you are trying to talk is either non-existent, or switched off, or busy. The printer server will not listen if it is already printing someone else's output: you will have to wait until it is ready

for you. If you get "Not listening" after a *I AM command, try logging on to the full file server number (usually *I AM 0.254 <user id.>). If this does not work then check with the person in charge of the network that the file server is running. The BBC Microcomputer expects the file server to be on Station Number 0.254 unless you tell it otherwise (some games and similar software can accidentally do that). If you wish to use a different file server, or to change back to file server 0.254, you will have to tell the computer explicitly. See Chapter 6 for details, also Section 4.4 for further information on network debugging.

Error A.3 (Decimal 163) - No Clock

There are three possible causes of this message. First, and most likely, the computer is not plugged into the network. Check that the Econet plug in the back of the computer is plugged in firmly, and that the other end of the lead is plugged into the wall socket or cable adaptor that connects to the network.

Second, there must be (only) one clock in the network. It is possible that it has been unplugged from the mains or from the network, or that there is a fault in the clock unit. Check that it is plugged in correctly. If this was not the problem, connect the clock unit to only one BBC Micro. Press N and BREAK as before on this machine. If the "No Clock" message still appears, suspect the clock unit.

Third, there is a fault on the network line itself. This is unlikely to happen if the network has functioned before, but could happen as a result of mechanical damage to the cable, or a fault in one of the machines attached to the network, or in one of the network terminators. Remove these items one by one until there is no "No Clock" message when you press N and BREAK. (Note that the message will not change on its own: you have to press N and BREAK).

Error A5 (Decimal 165) - No reply

The "No reply" message will occur if the filing operation fails in the middle, perhaps due to the file server being stopped in the middle of the operation.

4.2 File Server Errors reported by the BBC machine as Error A8 (Decimal 168)

FS Error 01 (Decimal 1) - Nothing happened

This error results from the use of a wildcard in a context where it may specify several files, *DELETE for example. Either there were no files which matched the filename, or errors occurred while processing those that did match. In the latter case, the individual errors can be found by repeating the operation with one of the files named explicitly.

FS Error 03 (Decimal 3) - Password file changed

A user has attempted to set a password or boot option, and the password file has been edited since the user logged on.

FS Error 04 (Decimal 4) - Bad number

A number has been incorrectly specified. The number may be in the wrong base, larger than the maximum value allowed, or simply not supplied at all.

FS Error 05 (Decimal 5) - Key locked

Only occurs if a system privileged user attempts to use a system privileged command when the Key is not in the "SYST" position. Only appropriate to HDFs.

FS Error 06 (Decimal 6) - Too short

If the system manager has set the appropriate flag in their password file entries, users are prevented from SAVEing files shorter than 16 bytes, in order to guard against the possibility of pressing BREAK before saving a program, for example. If it is necessary to create a short file, this may be done by means of OPENOUT and BPUT.

FS Error 07 (Decimal 7) - Circular RENAME

Whilst directories may be renamed into other directories, it is not possible to rename a directory into itself or one of its sub-directories as this would create a circular structure of directories.

FS Error 09 (Decimal 9) - Printer busy with station xxx

Generated by *PRINTOUT, if the printer is not free

FS Error 0A (Decimal 10) - Not authorised to use printer

Generated by *PRINTOUT, if the system manager has restricted the use of this printer to holders of a certain account (See § 3.5, EDITPRINT program)

FS Error 35 (Decimal 53) - File too big

Files may not exceed 8M bytes in length.

FS Error 46 (Decimal 70) - Illegal attribute

An attempt has been made to set an illegal combination of attribute bits . setting read or write access to a directory, for example.

FS Error 4F (Decimal 79) - Bad ARG to examine

The argument to one of the examine (catalogue) calls is out of range.

FS Error 85 (Decimal 133) — Not supported

The file server function code is not one of the recognised values.

FS Error 6D (Decimal 109) - Bad ARG to read arguments

The argument to the call to read random access information is out of range.

FS Error 90 (Decimal 144) - Bad time

The set time function has been called with a time or date that is invalid (eg. 32nd August).

4.3 The following File Server Errors are reported with their correct error numbers

Note that some of the errors have the same error number, despite having different text messages. This is to assist compatibility with Acorn and other systems -- the error number has been made the same as an existing one, when the action required (by a program, for example) to recover from the error would be similar. For example, **Disc full** and **Account xxx bankrupt** would typically require space to be made by deleting something, hence these both have error number C6.

Error A8 (Decimal 168)

"Catch-all" error number, see section 4.2 above.

Error AE (Decimal 174) - Not logged on

The specified user is not logged on to this file server. This error arises from utilities such as *NOTIFY FRED hello.

Error BO (Decimal 176) - Renaming across discs

It is not possible to transfer a file from one disc to another by use of *RENAME.

Error B3 (Decimal 179) - Directory full

A directory may not contain more than 255 entries.

Error B4 (Decimal 180) - Directory not empty

It is not possible to delete a directory unless it is empty (ie. contains no files or other directories).

Error B5 (Decimal 181) - xxx is a directory

It is not possible to LOAD a directory or to open it for output. While it is possible to open a directory for input, any attempt to use BGET will cause this error (see *ACCESS in chapter 6). The error may also be caused by an attempt to create a file of the same name as an existing directory.

Error B8 (Decimal 184) - Too many users

The file server has a maximum number of users (typically 60). The file server will not log stations off automatically, so it would be possible to log on, then at some stage change the station number. The file server user list would then contain a user logged on at a non-existent station. The solution is to stop and re-start the file server.

Error B9 (Decimal 185) - Bad password

Passwords have the same restrictions as file names concerning illegal characters. In addition, no wildcards are allowed.

Error BA (Decimal 186) - Insufficient Privilege

This error occurs when a user without system privilege attempts to perform some privileged operation, such as editing the password file. This error can also be caused by a user attempting to change his password or boot option, if this has been prohibited by the system manager.

Error BB (Decimal 187) - Wrong password

The password specified in the *I AM or *PASS command does not match that stored in the password file.

Error BC (Decimal 188) - User not known

If a user attempts to log on, and his user identifier does not appear in any password file in the system, and no default user has been set by the system manager, then this error will be produced. It is most commonly a result of mis-spelling the user identifier.

Error BD (Decimal 189) - Insufficient access

The user does not have sufficient access to the file or directory in question. If the main or auxiliary account numbers occur in the list of accounts associated with the user he has 'Owner' access, otherwise he has 'Public' access. Random access reading or writing of files is controlled by the W and R access attributes. Those before the stroke apply to owners and those after the stroke to public users (see *ACCESS in chapter 6).

To create a new file, the user must have owner access to the directory in which it is to be created.

To SAVE a file over one of the same name, owner access is required to the file and it must not be locked (L attribute), but the W and R attributes are not checked.

Users may check the access attributes and account numbers of files or directories by use of *INFO (the accounts are the last two items displayed), and may check their own list of accounts by use of the *STATEMENT utility.

Error BD (Decimal 189) - Not ENABLEed

The system manager can set an option to require a user to type ***ENABLE** before using *DELETE using a wild card. This is to prevent inexperienced users from deleting files accidentally. Once *ENABLE has been typed, its effect remains until the user logs off.

Error BE (Decimal 190) - xxx is not a directory

The name of a file has been specified in a context where a directory is required -for a catalogue or as the directory to search for a file (e.g. in a *DIR command).

Error BF (Decimal 191) - Who are you?

The user is not logged on to the file server, and should use the *I AM command to log on.

Error C0 (Decimal 192) - Too many files open

There are 8 channels available, of which two are permanently allocated to the Library and User Root Directory. In addition, the currently selected directory may require one, and one is required during the execution of the *DIR command. The remainder may be used for random access files.

Error C1 (Decimal 193) - File not open for update

The random access file in question has been opened for reading only (e.g. using OPENIN from BASIC), followed by an attempt to write to the file.

Error C2 (Decimal 194) - Already opened by xxxx

The file or directory has already been opened by the specified user (and possibly by others). It is not possible to delete files or directories which are open, nor to write to files. It is possible to read files which are already open, so long as they have been opened for input only. Note that each user is considered to have opened his User Root Directory, Library, and Currently Selected Directory.

Error C3 (Decimal 195) - Locked

An attempt has been made to DELETE, SAVE over, OPENOUT over or RENAME a file or directory which is locked (access L is set). Use *ACCESS <name> -L to unlock it.

Error C4 (Decimal 196) - Already Exists

An attempt to RENAME a file when a file already exists with the new name is not allowed.

Error C6 (Decimal 198) - Disc full

There is insufficient free space on the disc. Note that there may be unexpected overheads if saving a file requires a directory to be extended, and that a file longer than 16K incurs an overhead of 1K for each 512K of extent. It is possible to create files of large extent where the data blocks do not all exist - unwritten blocks will read back as zeros, and disc space must be found when they are first written. All disc space allocation is in units of 1K.

Error C6 (Decimal 198) - Account xx Bankrupt

The account to be charged for the current operation does not hold sufficient credit. Accounts are charged in the same way as disc space is allocated : see notes under 'Disc Full' concerning overheads. When a file is created, it is charged to the account of the directory in which it exists. If *ACCOUNT is used subsequently to change the account, the disc space taken by the file will be charged to the new account and re-credited to the old; the space taken by the directory entry is still charged to the account of the directory.

Error C7 (Decimal 199) - Disc Error

A media error occurred while trying to read the disc. Disc errors have serious consequences for the integrity of the system and should be reported to the system manager immediately, as they may indicate an impending problem with the disc or disc drive.

Error C8 (Decimal 200) - Disc changed

The disc has been changed. Log on again with *I AM. (This error is never generated by SJ File Servers, since "Who are you?" is more helpful.)

FS. Error C9 (Decimal 207) - Disc read only

On a floppy disc (or RM380Z) FS, this error means that a write protect tab has been applied to the disc, to which the user is attempting to write. (If the system manager is going to remove the tab, he must go through the normal disc changing procedure). On a hard disc FS, this error indicates that an on-line backup is in progress (during which it is not possible to write to the disc).

Error CC (Decimal 204) - Bad name

The filename used is illegal. Filenames may not contain the characters \$ % . ^ or : except where they are used for their special meanings - see Section 6.2. Characters above 126, control characters and spaces are not allowed at all.

Error CC (Decimal 204) - Bad wildcard

A wildcard character has been used where this is not allowed. Wildcards are not allowed in the filename used to create a file (although they may always appear in the directory name), or in passwords. This error will also occur if the number and types of wildcards do not match in the two filenames in a *RENAME command.

Error CF (Decimal 207) - Bad attribute

One of the attributes specified in a *ACCESS command is not one of the permitted attributes P D L W or R.

Error D6 (Decimal 214) - xxx not found

The specified file, disc or directory could not be found. Note that this error will be caused if you try to gain access to someone else's file which has been set to Private in the *ACCESS command.

Error DE (Decimal 222) - Channel

The channel number specified does not refer to a file which is open. This error can be caused by closing a file channel too soon, but the most common reason is that the 'context handles' have been lost in the client machine. These handles are required for almost all fileserver operations, and can be lost by switching off the computer, or using software which overwrites memory illegally (many games do this!), or due to various bugs in the Acorn NFS ROM. The solution is to log on again with *I AM xxx.

Error DF (Decimal 223) - EOF

An attempt has been made to read data beyond the end of a file.

Error FD (Decimal 253) - Bad String

The text string supplied is illegal, due to mis-matched quotes or an incomplete I sequence.

4.4 Fault-Finding an Econet Network

This Section contains a guide to finding faults on an Econet system.

If you are having problems which result in no communication at all, or only unreliable communication between computers on the network, then follow this procedure :

1. Unplug everything from the network, and make a very small network comprising one BBC Microcomputer, the Econet clock, a File Server and one terminator, with the necessary leads and adaptors to connect them all together. Open the clock box and turn on PERIOD switches 4 and 5 (marked 2 us and 4 us respectively) and turn on MARK switch 4 (marked 1 us). All other switches should be off.
2. Hold down letter **N** on the BBC Micro, and press and release break, then release **N**. There should appear at the top of the screen

BBC Microcomputer

Econet station nnn

BASIC

>

If the second line reads **Econet station nnn No clock**, then suspect the clock, but first try the same experiment with different connecting leads and a different BBC Micro.

3. If the BBC Micro does not give a **No clock** message, then try logging on to the File Server with the command ***I AM 0.254 <user id.>**. **If there is a pause of about a minute, followed by Not listening or Line jammed**, then continue on to step 4. If there is a prompt > after a short period, or an error message like **User not known** or **Incorrect password**, then you can conclude that the File Server and one BBC Microcomputer are functioning correctly, and proceed to step 7.

4. Reset the File Server station number to 254 as described in §3.3.3. (This step does not apply to RM380Z File Servers, but the Econet card in the RM380Z should be checked, that its station identifier links are correct.) Try logging on again as described in step 3 -- if this now functions, the File Server real-time clock had failed, either because the unit had not been switched on for more than about 6 months, or because of a hardware fault (contact your dealer or SJ Research in this case). Note that you will have to reset the time (and on the FDFS, the Baud rate) after this step. If the File Server now functions, proceed to step 7, otherwise continue.

5. Check the installation and operating instructions for the File Server to check that all has been done correctly. If you still cannot log on, then contact your dealer or SJ Research for advice.

6. It is still possible to check the network communication without a File Server. You will have to key in the following short program, which checks that the network and Econet interfaces function correctly. It can also be used to check an Acorn Level 2 File Server hardware, by stopping the program, then pressing <Ctrl-Break> on the File Server computer.

Key in this program (you do not of course need the REM statements) :

```
10 REM Program to test Econet  
20  
30 DIM X% 20
```

```

40 REPEAT
50 INPUT "Station to test : "stn%
60 Y% = X% DIV 256
70 ?X% = 1 : REM this is the reason code for 'send string'
80 X%?1 = stn% : X%?2 = 0 : REM X%?2 = network number for multiple networks
90 $(X%+3) = "I Are you there ??"
100 A% = &14
110 CALL &FFF1 : REM Osword
120 UNTIL FALSE

```

It would be wise to save this program, and also NETMON and STATIONS (if you have managed to get them off the File Server) on to local disc or cassette.

The program will repeatedly prompt for station numbers, and will execute the same call that *NOTIFY uses, sending string **Are you there ??** to each station specified. The target station must have its protection byte unset --_the easiest way to do this is to press <Ctrl-Break> on every station.

Start with only two BBC Microcomputers on a short network with just the clock, one terminator, and connecting leads and adaptors. Check that you can send the message from one to the other. Sending the message only one way in fact caused a bi-directional communication, so has fully tested both network interfaces.

Possible errors are (note that some of these errors only appear after the computer has been trying for about a minute) :

No clock	although you should have checked this already by pressing <N-Break> on each computer. Repeat step 2.
Net error Line jammed	First try unplugging the terminator, and replacing it with the other one. If this does not cure the problem, then try replacing the connecting leads one at a time. Finally, replace the BBC Micros one at a time. If you still cannot make contact between any two computers, check that you do not have any consistent errors : check that your connecting leads are wired pin 1 to pin 1 etc. (it is easiest to do this with a multimeter and an assistant), and ensure that whoever did the Econet upgrades to your BBC Micros ran the full test (using a special Econet tester). If the network has worked in the past, it is possible that a lightning strike or large transient has destroyed all the SN75159 Econet line driver chips -- replace one or two and try the test again.
Not listening	Check first that the protection byte is not set, by typing <Ctrl-Break> on the destination computer, and re-run the program. If it still fails, follow the instructions under Net error.

If some or all of the BBC Micros fail to communicate, then try new SN75159 Econet line driver chips (the chip should be in a socket). They can blow if there is a large transient on the network, such as that due to a local lightning strike.

To help avoid a repetition of this, S3 Research sell transient suppressor boxes to absorb transients of this type, and protect the computers.

7. If some or all of the BBC Micros function in the very short network, but not in the main network, then first check that the network connections are all correct. The

easiest way to do this is with a multimeter (set to an ohms range) and a special DIN plug wired with resistors as shown in Fig. 1 below.

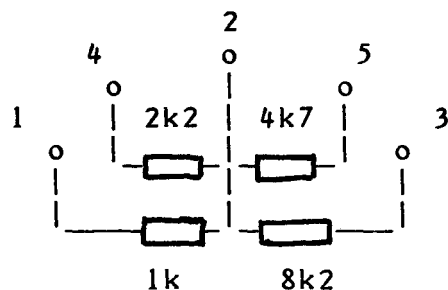


Fig 1 Test Plug (viewed from solder side)

Unplug the clock, terminators and all computers (including the File Server). Plug the test plug into a socket in the network. At each socket outlet on the network poke one of the prods into the centre pin (pin 2) of the DIN socket. Then check that the resistances are as follows :

- pin 1 to pin 2 1k
- pin 4 to pin 2 2k2
- pin 5 to pin 2 4k7
- pin 3 to pin 2 8k2

If any of these measured resistances differ significantly from these values, suspect an open or short circuit in the network. Remember that the network is split in the middle at the clock box, so the test plug will have to be plugged in on one side, and then on the other.

5. Basic Introduction to the Econet[®] for a User.

This chapter is still in preparation. For the time being, please refer to the Econet[®] User Guide published by Acorn Computers Ltd.

This chapter will be sent to you in due course.

6. Econet Client Commands

This Section gives a full description of the high level interface to the Econet system in BBC Microcomputer using the SJ Research (or Acorn) Econet system.

The chapter is split into several sections, as follows :

- 6.1 Explanation of Syntax Definitions
- 6.2 File, Directory and General specifiers
- 6.3 Wild Card specifiers, and rules for their use
- 6.4 Disc names
- 6.5 Printing through the network
- 6.6 Full descriptions of all commands (and utility programs)

Some of the commands described in section 6.6 are interpreted directly by the BBC Microcomputer operating system, the local NFS software, the BASIC language or the File Server, others are transient programs, which are executed in a designated area of system workspace, and will not corrupt any main program: for example the program ***TIME** or its variants will not affect the normal operation of a BASIC program.

Other programs in this Section are BASIC programs, and will overwrite any other program present -- but these are mostly utility programs which do not need to run with another program: for example COPIER which copies files from disc to network etc. They can of course be merged with other programs if desired.

The title Command Type and the suggested syntax will tell you which type of program or command each one is. The transient and utility programs are supplied in the system library (directory \$.LIBRARY as supplied). To use one of the utility programs in the library, type just the command, for example

***TIME**

There must not be a program with the same name in the currently selected directory. If there is such a file, it will be necessary to type

***\$.LIBRARY.TIME**

The mechanism of library searching is described under the *RUN or *LOAD commands.

In the case of transient or utility programs, there may be some described which are not available on your network system. This is because they would not be required for the type of work being done in your establishment, and so your System Manager will not have supplied them in the library.

6.1 Syntax Definitions

For each command, the syntax is given in Backus-Naur form with the extension that braces {} indicate optional repetition (or omission) of an item. Backus-Naur form is introduced in the BBC Microcomputer User Guide, Section 33.

The following notation is used :

::= means "is defined as"
| separates between alternatives
[] means that the enclosed item is optional
{ } means that the item type may be repeated or omitted
< > means that the enclosed item is a term defined elsewhere

Note that all "*" commands may be abbreviated in the BBC Microcomputer system, for example ***DE.** will be read as *DELETE

If there is any ambiguity arising from the use of a command or an abbreviation, these rules are followed:

The BBC Microcomputer Operating System will check its own table of commands, and then those of any language or filing system ROMs in the computer. Hence *D. will be read as *DISC (assuming disc system fitted), and not as *DELETE

The command will then be passed to the File Server to check its table of commands. If the command is not a File Server command, then the File Server will search first the currently selected directory (CSD), then the library, for a file of the same name as the command. The BBC Microcomputer will then *RUN this program.

If the command typed is ***<characters>**. (note the dot after <characters>) the file server will search the CSD, then the library, for the first match to <characters>* and *RUN this program. See section 6.3 on wild card specifiers for full details.

If you have a program which has the same name as one of the system commands (or you are not sure), use the *RUN command or its abbreviation */ to ensure that the program is run. An example of this is the program *TYPE -- computers which have a DFS fitted will execute *TYPE from ROM. In the File Server library, there is a version of *TYPE which runs much faster on the network. To be sure of using the right version, use

***/TYPE <file specifier>**

Note that the command *| (vertical bar) introduces a comment. This is useful in EXEC files, to introduce comments or messages.

Notes : *I. will be read as *INFO, but *I . -- note the space between I and the dot -- will be read by the local NFS as *I AM

Note also that *EX is a command in its own right, but that *EX. will be read as *EXEC by the operating system.

6.2 File, Directory and General Specifiers

A file specifier is defined as:

<file specifier> ::= [<directory specifier>.]<name>

where

<directory base> ::= :[<disc name>] | \$[<disc name>] | & | @ | ^

<directory specifier> ::= [<directory base>.]<name>.<name>

A general specifier is either a file specifier or a directory specifier, and is defined as

<general specifier> ::= <file specifier> | <directory specifier>

Each <name> is of maximum length 10 characters, alphanumeric, the characters ! - _(underscore)

or the wild card characters * or # (explained below). Upper and lower case characters are treated as equivalent, so that **File**, **FILE** and **file** all refer to the same file. Whilst some File Servers will accept other characters than those listed above, it is recommended that programmers use only those in the list, as this will assure compatibility between different versions of File Servers and from different manufacturers.

The last name specifies the file, and the previous one(s) directories. The directory specifier(s) and dot(s) may be omitted if the file is in the currently selected directory (see under *DIR for details) or in the current library (see under *LIB).

The <disc name> and the colon (or \$) may be replaced by \$ or : on its own if the file is on the currently selected disc.

Each user has a user root directory (URD), usually having the same name as the user identifier used with the *I AM command. This directory may contain files and other directories, and these latter directories may themselves contain further directories as well as files. For example:

```
*I AM FRED  
LOAD "SYSTEM.DEMO.PROG1"
```

loads a file called PROG1 in the directory DEMO. Directory DEMO is itself in SYSTEM, and SYSTEM is in the directory FRED.

There is a root directory on each disc, which is the directory containing all the user root directories. The name of this directory is the same as the name of the disc, so if the disc is called MAIN1, the full name of the file above is

```
:MAIN1.FRED.SYSTEM.DEMO.PROG1
```

This directory hierarchy is useful for keeping associated programs or text together. A number of commands can operate on complete directories, allowing time to be saved.

The following abbreviations are available:

\$ refers to the system root directory on the currently selected disc. In the above example \$ is equivalent to **\$MAIN1** (or **:MAIN1**)

: is an exact synonym of \$

^ refers to one level up the hierarchy. If user FRED had selected **SYSTEM.DEMO** as his current directory with the *DIR command, then ^.A1 is equivalent to

SYSTEM.A1

- & at refers to the user root directory. In the above example &.FILE2 would be equivalent to \$.
FRED.FILE2

- @ refers to the currently selected directory (CSD). @.PROG1 refers to the same file as PROG1 on its own, except that if PROG1 is not found in the CSD, the library will not be searched. In fact, this form would only be used if the programmer did not want the system to search the library for a file.

6.3 Wild Card Specifiers

A "wild card" in a file specifier allows reference to a group of files. The following two wild card characters are available:

- # matches any single character. Hence PROG# will refer to PROG1, prog2, and ProgC, but not PROG
- * matches any number of character, including zero characters. If there were a directory PROGS in the current directory, and it had three files "XYZ", "A1" and "A2" in it, then *DELETE PROGS.* will delete them all.
- . (dot) as the last character of the file specifier has exactly the same effect as * at the end, so that *DELETE P. will delete all files in the CSD that begin with letter P

The following rules apply to wild card characters:

For any operation, if a wild card is used in the directory name then the first directory (alphabetically) will be referred to only. For example, if directories A, B and C were in the current directory, then *DELETE *.* will be equivalent to *DELETE A.*

The last part (i.e. file name) in a file specifier may not contain a wild-card character in a SAVE, OPENOUT, *SAVE or *CDIR command.

In *DELETE, *ACCESS, *RENAME and *ACCOUNT the use of a wild card in the file name will cause the operation on all files or directories that match. The system manager can set an option for each user, which requires the user to type *ENABLE before a wild card delete operation. If this option is set, then the error **Not ENABLEed** will be given.

In *DIR, LOAD, *LOAD, *RUN and other commands the use of a wild card in the file name will refer to the first (alphabetically) of the possible files.

In a multiple match operation, if a filing error (e.g. insufficient access) would occur as a result of the requested operation, then that file (or directory) will be passed over, and the operation will continue on the rest. If it was not possible to do the operation on any file, then the error message **Nothing happened** will be displayed.

6.4 Disc Names

Each disc is given a name by the person in charge of the system when the disc is initialised. A disc name may be up to 10 characters, taken from the same characters as legal file names.

When referring to a file specifier in its full form, note that the colon : or dollar \$ must precede the disc name. This tells the system that the first part of the specifier is a disc name, and not a directory name.

Note that, for example

\$BOOT or **:FRED**

are disc names, whereas

\$.BOOT or **:.FRED**

are names of directories in the root of the currently selected disc.

6.5 Printing through the Network

In a network system, it is unlikely that individual users will have their own printers (except perhaps for printer devices for special applications).

The general purpose printer(s) will normally be connected to the network, via a printer server. A printer server is a station on the network, which will normally be the SJ Research File Server, or a BBC Microcomputer.

6.5.1 Selecting a Network Printer

The BBC Microcomputer assumes that printer output is to be sent to the local parallel printer, unless this is otherwise specified with the *FX5,n command : in particular *FX5,4 will select the network printer. Note that the BBC Micro will not allow you to change the printer selection, if there are still characters awaiting printing to the previous printer. If you need to throw away the contents of the printer buffer, type <escape><Ctrl-C><escape>

Initially the BBC Microcomputer will assume that the station number of the printer server is 235, and it will attempt to send output to this station if printing is requested.

The station number of the printer server may be changed using the *PS command, which will also select the network printer (*FX5,4 call). For example, *PS 236 will cause further output to be sent to station 236. If the station selected is busy, not running the printer server program, or non-existent, then the message **Not listening** will appear after a pause, when printing is attempted.

The command *PS on its own, or *PS followed by a name rather than a number, will have a different effect -- it will broadcast for a printer server and then display a series of lines of the form :

Printer nnn <status>

where <status> is one of **ready busy** or **offline** followed by

Printer nnn selected

*PS will attempt to select a printer answering **ready** for preference, followed by **busy**, then **offline**. If no suitable printer answered, then

No station responding

will be displayed.

*PS followed by a name will broadcast for a printer with that name. For example, a large network may have several Microline and several Diablo printers. The system manager will probably have given these printers names like ML and DIABLO respectively, and a user will be able to select one of the appropriate type by typing, for example :

***PS ML**

to select a Microline. The same output will be displayed as before.

There are in addition two reserved names for printers, AUTO and PRINT. For SJ Research File Servers that have two printers connected, the name AUTO will select printers automatically in a given order, as determined by the system manager. The File Server will keep a record of the selection made by each logged-on user, and will send further printout to this printer until the selection is changed.

Users not logged-on to the SJ Research File Server may not select a printer by name -- they will always get the AUTO printer. This is because the File Server cannot keep a printer type selection for anonymous users. In fact, the system manager may have set an option to deny use of a printer to anonymous users, in which case *PS will not find the printer(s) attached to the File Server unless the user has logged-on.

Full information is given under the *PS command in Section 6.6.

6.5.2 Turning the Printer On and Off

To start printing the user should print the character <Ctrl-B>, either by typing it, or by **VDU 2** from a program. The printer will usually be set up by your system manager to print a banner, which will be a series of characters designed to be easily visible, followed normally by the user's name, the station number, time and date.

The characters sent to the printer will then be printed. The BBC Microcomputer always filters one character value out -- by default this is character &0A (line feed), so that printers that do an automatic line feed after every carriage return can be used. To change the character that is filtered, type ***FX6,<character>**. If the printer does not have auto line feed, then ***FX6,0** is a useful choice. The program *PS will perform this *FX6,0 automatically if the printer(s) in your network require it (the system manager will configure *PS to match the printer).

Printing is concluded by printing <Ctrl-C> (**VDU 3**) at the end of the printout. The printer will usually be set up to leave a blank page, and will then be ready for another user's output. Until the current user types <Ctrl-C>, the printer server will remain busy with this user's output unless its internal timer finishes the user's printout after a sufficient period of inactivity. Note that this if the printer server times out in this way, some of the characters sent to the printer will be lost.

To stop printing temporarily, do not use <Ctrl-B> and <Ctrl-C>, since this will result in the print job being ended, thn another one begun on a fresh page with a fresh banner. Use ***FX3,<number>** to do this, where <number> is a single byte with each bit having the following effect :

Bit 0 (1)	1=enable RS423 (serial) output
Bit 1 (2)	1=disable output to VDU
Bit 2 (4)	1=disable printer output
Bit 3 (8)	1=enable printer (but it must have been started with <Ctrl-B>)
Bit 4 (16)	1=disable output to any *SPOOL file
Bit 5 (32)	no effect
Bit 6 (64)	1=disable printer output, except for characters preceded by <Ctrl-A
Bit 7 (128)	no effect

For example, calling *FX3 with bits 3 and 1 set to 1's (*FX3,10) will disable the VDU and enable the printer, i.e. will print only. Calling with bit 2 = 1 (*FX3,4) will send output to the VDU only (even if <Ctrl-B> has been typed). Calling with no bits set (*FX3,0) will send output to both VDU and printer (this is the default setting).

For printing graphics to the printer only (where it is important that all characters, including the one specified in the *FX6 call, are printed), use the following in your program .

VDU2 start the printer
***FX3,0** unnecessary unless you have previously called *FX3
VDU1,<character>

VDU1 (<Ctrl-A>) sends the immediately following character to the printer only, but the VDU output must be active for it to work (!) The character following <Ctrl-A> will be sent to the printer, regardless of whether it is the character filtered out (set by the *FX6 call -- see above)

For programs that do a lot of switching between printer only and VDU only, the following calls are recommended :

VDU2 start the printer
***FX3,64** this disables printer, except for chars preceded by VDU1
VDU <character> or PRINT <anything> sends to the VDU only
VDU1,<character> sends to the printer only

This saves multiple vast numbers of *FX3 calls if there is a lot of switching between VDU and printer.

Please note that the documentation of *FX3 in the Advanced User Guide for the BBC Micro, page 119, is misleading. Line 7: in Econet it is always necessary to send <Ctrl-B> to start printing. Lines 18-20: untrue. You may use Osbyte &EC to read the current state of the bits, but Y will not work as a bit mask for Osbyte 3 !

Please use this *FX3 call rather than using VDU21 and VDU6 to turn the screen on and off. There are bugs in this part of the VDU driver which will cause undesirable results.

6.5.3 Off-Line Printing from the File Server

There is a facility available to print the contents of a file off-line, if desired. The command is *PRINTOUT, and there is a corresponding command *PRINTER to select the printer at which the output appears.

There are two advantages to the use of *PRINTOUT. First, the job can proceed without using any processing power of the BBC Microcomputer. Second, the system is not restricted by the defined printer protocol, and can therefore give much more information back to the user.

See the full description of these commands in the next Section.

6.6 Command Descriptions

This section gives a full description of the commands and utility programs for the Econet system on a BBC Microcomputer. The descriptions are arranged alphabetically.

The commands described are as follows

*ACCESS	File Server command
*ACCOUNT	File Server command
BPUT#	BASIC keyword
BGET#	BASIC keyword
PTR#	BASIC keyword
EXT#	BASIC keyword
EOF#	BASIC keyword
INPUT#	BASIC keyword
PRINT#	BASIC keyword
CLOSE#	BASIC keyword
*BUILD	Transient program
*BYE	N.F.S. command
*CAT	BBC Micro O.S. command
*CATALL	Transient program
*CDI R	File Server command
*CLOSE	Transient program
COPIER	BASIC program
*CV	Transient program
*DEFACCESS	File Server command
*DELETE	File Server command
*DIR	File Server command
*DISABLE	File Server command
*DISCS	Transient program
*DUMP	Transient program
*ENABLE	File Server command
ERAQ	BASIC program
*EX	N.F.S. command
*EXEC	BBC Micro O.S. command
*FREE	Transient program
*FSLIST	Transient program
*GO	Transient program
*GTIME	Transient program
*GUSER	Transient program
*I AM	N.F.S. command
*INFO	File Server command
*LIB	File Server command
LOAD	BASIC command
*LOAD	BBC Micro O.S. command
*LOGON	Transient program
MULTICOPY	BASIC program
*NOTIFY	Transient program
OPENIN	BASIC keyword
OPENUP	BASIC 2_keyword
OPENOUT	BASIC keyword
*OPT1	BBC Micro O.S. command
*OPT4	BBC Micro O.S. command
OSCLI	BASIC 2 keyword
*PASS	File Server command
*PRINT	Transient program
*PRINTER	File Server command
*PRINTOUT	File Server command
*PS	Transient program
*PTIME	Transient program
*PDATE	Transient program
*PDATE2	Transient program

*PUTGET	Machine code program
*PUTGET2	Machine code program
*RENAME	File Server command
*RUN	BBC Micro O.S. command
SAVE	BASIC command
*SAVE	BBC Micro O.S. command
*SDISC	File Server command
*SPOOL	BBC Micro O.S. command
*STATEMENT	Transient program
*STATIONS	Transient program
*TIME	Transient program
*TYPE	Transient program
*USERS	Transient program
*VERS	Transient program

*ACCESS

Command Type : File Server command

Syntax :

***ACCESS <general specifier> <access string>**

Action with Wild Cards in the File Name : Occurs on every match. Note the special use of access letter D below.

Description :

This command allows the access status of a file to be changed. New files are created with default access status WR/R, unless this is changed using the *DEFACCESS command (q.v.).

An owner is defined as someone with access to the account or the auxiliary account of a file (or directory). A non-owner has public access to a file (or directory).

The access string is of the form

[<owner access>][/<public access>]

where <owner access> and <public access> are strings of letters taken from the list below. If no characters appear in either string, then this signifies no access to the file (or directory) for that category of user. If no / appears, then all letters will be taken as defining 'owner' access, and 'public' access will be none.

Letters that can appear in either 'owner' or 'public' strings

No letter: No access to file until status has been changed

- R Read only access. The file can be loaded, or opened for reading using OPENIN (q.v. this Section).
- WR Read and write access allowed. This means that the file can be loaded, or opened for update using OPENUP (q.v. this Section)
- W Write only access. this means 'append to end only' -- i.e. The file can be opened using OPENUP, and written to, but only if PTR# EXT#. An example of its use is keeping a record of users of a program, without the users being able to read it.

Letters that describe general attributes of the item

- L Locked item: cannot be deleted or renamed until status is changed. The public can never delete or rename a file, so this applies only to the owner(s). A directory cannot be deleted until it contains no entries, so locking a directory is likely to be useful only to prevent renaming.
- P Private item: invisible to anyone but the owner. If a non-owner attempts to look at the appropriate directory using *CAT or *EX, these items will be listed as **...Private**, and if he attempts to perform any operation on the item, the error message **Not found** will be given.
- D Item is a directory: An attempt to change this access letter will cause an Error 46 (see below), but it may be used to specify directories in wild card operations -- see below.

Additional characters that may appear in the access string

- + adds the following letters to the existing access status
- subtracts the following letters from the existing access status.
- / separates owner access status (before the /) from public access status (after the /). There may not be more than one / in the access string.

The letters **L P** and **D** may be specified before or after the /, but will appear before it when the file is listed in a *CAT or *EX command. There may be any number of + or - signs in the access string.

Wild cards are permitted in the file specifier, and will cause the command to apply to all matching files (or directories). The usual rule concerning wild cards applies --

the operation will be applied only to files (or directories) where an error would not be caused. For example

***ACCESS Data* D+P**

will make private all directories beginning with the letters DATA, but not change any files (since the attempt to give the access letter **D** to a file would cause an error). There is however in this command an important extension to this rule, namely that **-D** can be used to specify a file not a directory. The command

***ACCESS Data* -D+ / W**

will give write access to non-owners, of all files (only) beginning with the letters DATA.

Note that, to change the access letters of the users own user root directory, it will be necessary to type (for example) :

***ACCESS ^.FRED +P**

Random Access Operations

The effect of the access status on random access operations is shown in the tables below:

Operation			
Access Status	OPENIN	OPENUP	OPENOUT
D	D	error IAD	error IAD
none	error IA	error IA	WR
W	error IA	W	WR
R	R	error IA	WR
WR	R	WR	WR

Table 1. Effective access after different file opening commands.

Operation	BGET#	BPUT#	Set/Read PTR#, Read EOF#
Effective Access			
D	error IAD	error IAD	error IAD
W	error IA	error IA unless PTR# = EXT#	OK
R	OK	error NOFU	OK
W R	OK	OK	OK

Table 2. Operations allowed for each type of effective access.

Error IA is the error message "Insufficient Access" (Error BD)
 Error IAD is the error message "xxx is a Directory" (Error B5)
 Error NOFU is the error message "File not open for update" (Error CI)

Examples :

***ACCESS Prog22 WR/**

will change the access status of file Prog22 to read and write for the owner, but no access at all for non-owners

***ACCESS Prog* +L**

will find every file (or directory) beginning with the letters PROG, and will lock each one.

***ACCESS * D+P**

will find every directory within the currently selected directory and add the letter P to their access status, i.e. making them all private access.

***ACCESS * -D+P**

will find every file within the currently selected directory and add the letter P to their access status, i.e. making them all private access.

Likely Errors :

FS Unusual Error 46 (70) -- an attempt to set an illegal attribute (e.g. W to a directory or D to a file)
 Error BD (189) Insufficient Access -- only an owner can change the access status of a file
 Error CF (207) Bad attribute -- attempt to use letters other than P D W L R

Compatibility Notes :

This command is supported by Acorn systems, but access letter **P** is not, nor are the + and - characters. Note also that Acorn systems define file ownership differently, and that there is no *DEFACCESS command.

***ACCOUNT**

Command Type : File Server command

Syntax :

***ACCOUNT <file specifier> [<main account no.>] [[<aux account no.>]]**

Action with Wild Cards in the File Name : Occurs on every match.

Description :

This command changes the main account number of the file specified to <account no.> and/or the auxiliary account number to <aux account no.>. Account numbers are two digit hexadecimal numbers, ranging between 0 and FF.

When a file is initially created, it is given the same main account number and auxiliary account number as that of the directory that it is in, and the space taken by the file is debited from the balance in the main account. As explained in Section 5, a user has owner access to a file if he has access to either the main or the auxiliary account, otherwise he has only public access.

When the main account number is changed, this command debits the space taken by the file from the 'new' account, and credits the same amount to the 'old' account.

The user must be an owner of the file, i.e. he must have access to either the main account or the auxiliary account of the file (or directory). If he is changing the main account number, he must also have access to the new main account. The auxiliary account may be changed to any value without restriction.

It is possible for a user with access to the auxiliary account only, to use this command to change either the main account number (in which case he must have access to the new main account) or the auxiliary account number (to any value desired). By the former action, this user can 'take over' the cost from the original creator of the file, and may remove the original creator's owner access as well. By the latter, he can transfer his own owner access to someone else, by changing the auxiliary account number to an account that someone else has access to.

Examples :

***ACCOUNT NewDump 25**

changes the account number of NewDump to 25. To do this, the user must have access to both the original account, and also account 25.

***ACCOUNT DumpProg (37)**

If user JOE had access to account 37, then this command would give Joe (as well as this user) owner access to the file DumpProg. This could be useful for several users all working on the same set of files.

***ACCOUNT New* 25 (37)**

This command changes all files beginning with the letters NEW to account number 25, and auxiliary account 37 (as above).

Likely Errors :

Error BD (189) Insufficient Access -- only an owner can change the account of a file
Error C6 (198) Account nn bankrupt -- There must be sufficient credit in the new account.

Compatibility Notes :

This command does not exist in Acorn systems, which do not support space accounting. Acorn systems use a different system to determine file ownership.

BPUT#, BGET#, PTR#, EXT#, EOF#, INPUT#, PRINT#, CLOSE#

Command Type : BASIC keywords

Syntax :

See the BBC Microcomputer User Guide, Section 33.

Description :

These commands are detailed in the BBC Micro User Guide in the section containing descriptions of all keywords alphabetically. The effects of using the BPUT#, BGET#, INPUT# and PRINT# commands when the access to a file is limited, is explained under the *ACCESS command (q.v.).

Note that the use of BPUT#, BGET#, INPUT# and PRINT# is very slow over the network from a BBC Microcomputer. The reason for this is that BASIC (and some other languages) sends bytes one at a time, requiring a complete network transaction (about 50 bytes sent in total plus file server overheads) for each single byte of useful information.

If there is much string fetching and putting to be done, the use of the OSGBP machine code call (See Chapter 7) is recommended. An easy way to do this is to run the transient program *PUTGET before beginning the session -- this plants code to convert single byte filing operations into the appropriate block operation: see the Section on *PUTGET in this chapter.

Users may alternatively wish to use their own call to the OSGBP routine, especially if they find the restrictions on the use of PUTGET make it unsuitable .

Procedure to call OSGBP from BASIC

At the head of the program allocate workspace,

```
DIM gb% 12           this is space for the arguments to OSGBP
```

and space for a buffer if required, e.g.,

```
DIM BUFF% 500      (for writing, this could alternatively be a string)
```

followed by, for example,

```
PROCgbpb(3,CHAN%,BUFF%,100,2000)
```

with this definition after the end of the main program,

```
DEFPROCgbpb(A%,channel%,buffer%,length%,offset%)  
LOCAL X%,Y%  
X%=gb%  
Y%=gb% DIV 256  
?X%=channel%  
X%!1=buffer%  
X%!5=length%  
X%!9=offset%  
CALL &FFD1  
ENDPROC
```

This example would load 100 bytes from offset 2000 in the file whose channel number is channel%, to the location given by the value of BUFF%

Other values of A% will have the following effects :

A%=1	Write bytes from buffer% to file, at offset% bytes from start of file
A%=2	Write bytes from buffer% to file, at current value of PTR#
A%=3	Read bytes from file to buffer% , at offset% bytes from start of file
A%=4	Read bytes from file to buffer% , at current value of PTR#

A%=5 to A%=8 other functions (See §8)

Likely Errors :

Error B5 (181) xxxx is a directory -- A directory may be opened using OPENIN (only), but an attempt to use BPUT# or BGET#, or to set or read PTR# or EXT# will give this error.

Error BD (189) Insufficient Access -- An attempt to read a file to which the user has only write access, or to write to a file to which the user has only read access.

Error C1 (193) File not open for update -- an attempt to write to a file that has been opened with OPENIN, or to write to a file with access W only with PTR# not equal to EXT#.

Error DE (222) Channel -- most likely because the file has not yet been opened, because BREAK has been pressed, or because the File Server has been restarted or the discs have been changed. Log on again (after BREAK or restart only !), and use OPENIN, OPENUP or OPENOUT as appropriate to open the file before using one of these operations.

Error DF (223) EOF -- after an attempt to read data beyond the end of the file.

Compatibility Notes :

All these keywords supported by Acorn systems. There are some small differences in the detailed interpretation. The use of *PUTGET is also recommended with Acorn File Servers.

***BUILD**

Command Type : Transient program

Syntax :

***BUILD <file specifier>**

Action with Wild Cards in the File Name : Wild cards prohibited.

Description :

This program creates a new file of the name <file specifier>, and then prompts for keyboard input, which is sent directly to the file. To end the file press the <escape> key. The file will be closed, and control returned to the current language.

A common use of this program is to create !BOOT (see under *OPT4, this Section) or other files which are going to be used -with *EXEC, although any text file may be entered using *BUILD.

This program opens a new file using the OPENOUT call, and then writes to the file using the multiple byte transfer operation OSGBP. It will therefore run much faster than, for example, the version of *BUILD contained in the DFS ROM. On a machine equipped with DFS, this program should be run by typing ***/BUILD** (see under *RUN for details).

Examples :

***BUILD !BOOT**

```
0001..... (type
0002..... your
0003..... text
0004..... here)
0005 <escape>
```

Likely Errors :

There are no errors specific to this program, but it opens the file using OPENOUT, so it can cause the same errors as OPENOUT (q.v.)

Compatibility Notes :

Supported on Acorn systems.

***BYE**

Command Type : Network Filing System command

Syntax : ***BYE**

Description :

Logs off from the currently selected File Server. The user's name and machine number are cleared from the current user list within the File Server. In addition the user's currently selected directory (CSD), user root directory (URD) and Library directory are de-selected, and any open files are closed.

It is recommended that all users use the *BYE command at the end of a session -otherwise someone else using this computer later on will have access to all this user's files and accounts. This is especially important for system users or others with access to special information.

Some applications programs may log on to multiple File Servers, and use the appropriate Osword calls to select between them. In this case *BYE will only log off from the most recently selected of these.

Likely Errors :

Error BF (191) Who are you ? -- if the user was not logged on. This error will also be produced if a filing operation is attempted after the *BYE command.

Compatibility Notes :

Supported by Acorn systems.

*CAT

Command Type : BBC Microcomputer operating system command

Syntax :

***CAT [<directory specifier>] | *.[<directory specifier>]**

Action with Wild Cards in the File Name : Occurs on first match only (alphabetically)

Description :

Although all File Server commands may be abbreviated with a dot at the end, it is worth noting that the minimum abbreviation of this command is *.

This command causes a list to be printed of the contents of the directory <directory specifier>, or of the currently selected directory (CSD) if the specifier is omitted. The form of the list is:

<directory name><seq. no.>	<access status>
<currently selected disc>	Option <option number>
Dir.<currently selected directory>	Lib.<currently selected library>
{<file name> <access letters>}	

The <directory name> is that of the directory being displayed, with a sequence number <seq. no.>, which is incremented every time a change is made to the directory. The <access status> is either **Owner** or **Public**, depending on whether the user has access to the account number of the directory. The name of the currently selected disc (see under *SDISC) is also displayed.

The <option number> is that set up by the user using the *OPT4 command (q.v.). The <disc name> is the name of the currently selected disc. The currently selected library (usually \$.LIBRARY, but see under the *LIB command) will appear after Lib: and the currently selected directory after **Dir.**

There then follows a list of the files in the directory, with their access status (see under the *ACCESS command). Some of the access letters (either before or after the / character) will be upper case, and the others will be lower case. The upper case letters indicate the access that this user actually has to this item. For example, access letters **WR/r** means that the user is an owner of the file, and can read it or write to it. If the access were **wr/R**, the user would have public access only to the file, and would be able to read it only. If the owner has set an item to access P, then it will appear only as **...Private** to a non-owner.

Examples :

*,

lists the currently selected directory, and will produce output similar to :

John (038)		Owner	
MAIN-1		Option 03 (exec)	
Dir. John		Lib. LIBRARY	
ABC	W R/r	File23	W R/r
...			
...			
*. PROGS			

lists the contents of directory PROGS, which itself is in the currently selected directory, and will produce output similar to :

PROGS (038)
MAIN-1
Dir. John

ADDER **WR/r**
...
...

Owner
Option 03 (exec)
Lib. LIBRARY
ScreenDump W R/r

***CAT \$.FRED.DATA**

lists the contents of directory DATA, itself in directory FRED, which is in the root of the currently selected disc:

***. :MAIN2.JOHN.PROGRAMS**

lists directory \$.JOHN.PROGRAMS on disc MAIN2 (which may or may not be the currently selected disc).

Likely Errors :

Error BE (190) xxxx is not a directory -- if the specifier after the command is the name of a file.

Error D6 (214) xxxx not found -- if the directory specified cannot be found.

Compatibility Notes :

Supported by Acorn systems -- but the access letters all appear in upper case on Acorn systems.

***CATALL**

Command Type : Transient program

Syntax : ***CATALL**

Action with Wild Cards in the File Name : Not applicable.

Description :

*CATALL produces a catalogue of the currently selected directory, and of all sub-directories within the CSD. The display is of the form :

```
<file name>
<file name>
Directory <directory name>
  <file name>
  <file name>
  Directory <directory name>
    <file name>
    .
    .
    .
<file name>
Directory <directory name>
  <file name>
  .
  .
  .
```

The file and directory names in the left-most column are those in the currently selected directory (CSD). Each time a sub-directory is encountered, its contents are listed, indented four spaces to the right. In this way a complete tree of directories is displayed.

*CATALL does not take an argument, so to list the contents of a different directory type,

```
*DIR <directory name>
*CATALL
```

Examples

```
*CATALL
```

```
!BOOT
!MAIL
BT Wiring
contnts
Ecommands
```

```
Directory Letters
  ADDRESSES
  Base
```

```
Directory Bloggs
  12-7-85
  13-5-85
  cougar 3
```

```
Directory Customers
  Alpha
  Bravo
  Charlie
```

Pers
RR

PokerBoot

Directory ReturnNote

 general
 Grommet
StEdsEP
ur2

Files !BOOT, !MAIL, down to PokerBoot, StEdsEP, ur2 are in the CSD. Letters, ReturnNote are sub-directories of the CSD, and Bloggs, Customers are sub-directories of Letters.

Likely Errors :

There are no errors specific to this command

Compatibility Notes :

Supported by Acorn systems.

***CDIR**

Command Type : File Server command

Syntax :

***CDIR <directory specifier>**

Action with Wild Cards in the File Name : Wild cards prohibited

Description :

This command creates a directory of that name. Note that if a directory of that name already exists, no action will be taken (and no error message will be produced). The user must have access to the account, or the auxiliary account, of the directory in which he is creating this new directory: but he may then change the account of the new directory to any other to which he has access.

Examples :

***CDIR PROGRAMS**

creates a directory called PROGRAMS in the currently selected directory

***CDIR \$.PROJECT.NEWDATA**

creates a directory NEWDATA in the directory \$.PROJECT. The directory PROJECT must already exist, and the user must have access to the account or the auxiliary account of the directory PROJECT.

Likely Errors :

Error CC (204) Bad wildcard -- an attempt to use a wild card character * or # in the new directory name.

Error C6 (198) Account nn bankrupt -- There must be sufficient credit in the account of the directory in which the user is creating this new directory.

Error BD (189) Insufficient access -- if the user does not have access to the main or auxiliary account of the directory in which the user is creating this new directory.

Error BE xxxx is not a directory -- if a file (not a directory) of that name already exists.

Error D6 (214) xxxx not found -- if the directory into which the new one is being created is not found. If, in the second example, the directory \$.PROJECT did not exist, the error **PROJECT not found** would occur.

Compatibility Notes :

Supported by Acorn systems -- except that Acorn systems do not support accounts, but use a different system to determine ownership. Note that Acorn File Servers will give an error after an attempt to create a directory if a non-empty (or locked) one of that name already exists. For compatibility, users' programs that use *CDIR should use the OSFILE call with A=5, which will check for the existence of the item (will return A=0 if there was no file or directory or file with the name specified, A=1 if a file of that name was found, A=2 if a directory was found). The important feature of this call, is that the library will not be searched for a directory (though it will be, for a file)

.

***CLOSE**

Command Type : Transient program

Syntax :

***CLOSE**

Action with Wild Cards in the File Name : Not applicable.

Description :

This program closes all currently open files. It is exactly equivalent to the BASIC command **CLOSE#0**, but can be run from any language or other system.

*CLOSE runs in Page 0 of the BBC Microcomputer, and so will not corrupt other transient programs.

Examples

***CLOSE**

Likely Errors :

There are no errors specific to this command

Compatibility Notes :

Supported by Acorn systems.

COPIER

Command Type : BASIC program.

Syntax :

CHAIN "COPIER"

Action with Wild Cards in the specified file names : Occurs on first match only (alphabetically).

Description :

This program is a general purpose copying utility, which will copy files of any size, between any two filing systems, or between two places on the same filing system.

The program will prompt :

Source filing system : *

Dest. filing system : *

To copy between two different filing systems, enter the names of the systems (e.g. DISC and NET) after the appropriate prompts. To copy between two directories, enter *DIR <directory name> after each, and similarly to copy between File Servers, type *I AM <file server station no.><user id.> after each.

The program will then prompt for the file name, and for the new name required in the destination filing system. Typing <return> after the latter will give the same name to both.

COPIER will *LOAD and then *SAVE the file if it is sufficiently small, otherwise it will call OPENOUT to create the new file, and will then read blocks of data from the source and write them to the destination, using OSGBPB calls.

* commands may be typed after either of these prompts, but note that, where a command applies specifically to one of the filing systems, it will apply to the source system if typed after **File name**, and to the destination system if typed after **New name**.

If a file is to be moved within one disc on a File Server, note that the *RENAME command (q.v.) will do this, and it is not necessary to use COPIER.

Where it is necessary to copy multiple files between File Servers, the utility MULTICOPY is provided. MULTICOPY will copy entire directory trees automatically.

Examples :

After **CHAIN "COPIER"**, the commands :

Source filing system : *DISC

Dest. filing system : *NET

will copy from floppy disc to network File Server. If it is necessary to select a directory (for example), this can be done with the *DIR command after the **New name** prompt.

Source filing system : *I AM 253 FRED

Dest. filing system : *I AM 254 FRED

will copy selected files between File Servers, probably on behalf of user FRED. After the file name prompts, it is possible to enter a full file specifier (e.g. \$.JOHN.BBCPROGS.THING). To save files the user must of course be an owner of the destination directory.

Source filing system : *DIR \$.JOHN

Dest. filing system : *DIR \$.FRED

This will copy selected files from \$.JOHN to \$.FRED on the same File Server. This is useful if FRED needs his own copies of the files (if for example he was likely to change his copy). If this is not the case, use *RENAME.

Likely Errors :

This program will respond **Come again ?** if the file is not found. Otherwise the errors produced by *LOAD, *SAVE, OPENIN and OPENOUT can occur when running this program.

Compatibility Notes :

Supported by Acorn systems.

***CV**

Command Type : Transient Utility program

Syntax : ***CV**

Action with Wild Cards : Not applicable.

Description :

This program displays the station numbers of the currently selected File Server and Printer Server, the users own station number, and the version of OSARGS in use. If there are multiple networks joined by bridges, then the network number will be given before the user's station number, in the form **<network number>.<station number>**

The OSARGS version number is the value that would be returned in A by the OSARGS call with Y=0 and A=2. This number depends on the local NFS version number :

OSARGS version no. 2 - NFS 3.34

OSARGS version no. 1 - NFS 3.6 or Advanced NFS

Examples :

***CV**

File server number : 254

Printer server number : 235

You are : 001.005

OSARGS version no. 001

Likely Errors :

The message **RxCB ?** will be displayed if no receive control block is available in the BBC Microcomputer.

There are no other errors specific to this program.

Compatibility Notes :

Supported by Acorn systems

***DEFACCESS**

Command Type : File Server command

Syntax :

***DEFACCESS <access string>**

Action with Wild Cards in the File Name : Not applicable.

Description :

This command applies to the currently selected directory. The string will be applied as the default access string to any new files created within this directory, or within any subsequently created subdirectory (unless DEFACCESS is used again to apply to that subdirectory).

For a list of the possible access letters, see the *ACCESS command. The default access for a directory will be listed by the *EX or *INFO commands (q.v.)

The user must be an owner of the currently selected directory.

Examples :

***DEFACCESS WR/R**

The root directory \$ on a new File Server disc has this default access, so directories in \$ will have this default unless it is explicitly changed:

***DEFACCESS PWR/**

will cause subsequently created files in this directory to have access letters PWR/ and subsequently created sub-directories to have access **PD/**

Likely Errors :

Error BD (189) Insufficient Access -- only an owner can change the default access status of a directory

Error CF (207) Bad attribute -- attempt to use letters other than P D W L R

Compatibility Notes :

This command is not supported in Acorn systems. The defaults are **WR/** for files, and **DL/** for directories on Acorn File Servers.

***DELETE**

Command Type : File Server command

Syntax :

***DELETE <general specifier>**

Action with Wild Cards in the File Name : Occurs on every match. The command ***ENABLE** will be needed before a wild card delete if the system manager has set the appropriate option.

Description :

This command deletes the specified file. The user must be an owner of the file, and the file must not be locked (access letter L).

When used with a wild card specifier, this command will delete every item that matches, within the specified, or currently selected, directory. The system manager can set an option to require the command ***ENABLE** before any wild card delete operation. If it is required to delete a tree of directories and sub-directories, the program ERAQ (q.v.) should be used.

It is only possible to delete a directory if it contains no entries.

Examples :

***DELETE VATPROG**

deletes the file (or directory) VATPROG in the currently selected directory.

***ENABLE**

DELETE *.

finds the first directory (alphabetically) in the currently selected directory, and deletes all items within this first directory (except ones where the attempt would cause an error). This is standard action with wild card specifiers, and is explained fully in Section 6.3 at the beginning of this chapter.

***DE. \$.JOHN.MYPROG**

deletes the file MYPROG in directory \$.JOHN. The user must be an owner of (i.e. have access to the main or auxiliary account of) the file MYPROG

There is also a command ***DISABLE**, which has the opposite effect of ***ENABLE** (both are described fully in this Section).

Likely Errors :

Error B4 (180) Directory not empty -- caused by an attempt to delete a directory that still contains some entries.

Error BD (189) Insufficient access -- caused if the user is not an owner (i.e. does not have access to the main or auxiliary account) of the item.

Error BD (189) Not ENABLEed -- If the system manager has set the ***ENABLE required** option for this user, the **Not ENABLEed** error will be given.

Error C2 (194) Already opened by xxxx -- If another user has this file open for reading or writing, then it cannot be deleted until it has been closed.

Error C3 (195) Entry locked -- It will be necessary to use the ***ACCESS** command to unlock the file before it can be deleted.

Compatibility Notes :

Supported by Acorn systems, but note that wild cards are not allowed.

***DIR**

Command Type : File Server command

Syntax :

***DIR [<directory specifier>]**

Action with Wild Cards in the File Name : Occurs on first match (alphabetically) only.

Description :

This command changes the currently selected directory (CSD) to the one specified. If it is used without a specifier, it will select the user root directory (URD) as the CSD.

When a directory is selected by this command, then it is deemed to be open by the File Server. The user root directory (URD) and library directory are opened at log-on, and remain open throughout the session. Hence the number of channels available for other filing operations is reduced by two from the maximum of eight if the *DIR command has not been used, and by a further one if *DIR has been used to select a different directory as the CSD. In addition, *DIR opens the new directory before it closes the old one (if appropriate), giving an instantaneous total of four directories open.

Examples :

***DIR PROGRAMS**

searches the currently selected directory for a directory called PROGRAMS, and makes that the new currently selected directory,

***DIR \$.FRED.LETTERS**

selects the directory LETTERS in directory \$.FRED as the CSD.

***DIR**

selects the user root directory as CSD.

***DIR ^**

selects the next directory up the tree (***DIR ^.^** selects two directories up) as the CSD.

Likely Errors :

Error D6 (214) xxxx not found -- if the specified directory could not be found.

Error C0 (192) Too many files open -- If several files are open for random access, it is possible that there may not be a channel left for the CSD.

Error BE (190) xxxx is not a directory -- caused by an attempt to specify a file as the CSD.

Compatibility Notes :

Supported by Acorn systems, but wild cards and the character ^ are not allowed.

***DISABLE**

Command Type : File Server command

Syntax :

***DISABLE | *DISABLE SAVES | *DISABLE LIBRARY**

Action with Wild Cards in the File Name : Not applicable.

Description :

This command is the exact opposite of the *ENABLE command (q.v. this Section).

***DISABLE** without a parameter prevents the use of *DELETE with a wild-card thereafter, until *ENABLE is typed. This command may be run automatically at log-on, depending upon an option set for a user in the password file, by the system manager.

***DISABLE SAVES** prevents a file of less than 16 bytes in length being saved. This command may be run automatically at log-on, depending upon an option set for a user in the password file, by the system manager.

***DISABLE LIBRARY** reduces the library searching facility provided in S3 File Servers, to a level equivalent to that in Acorn systems. That is to say, the library is searched only for ***RUN** or * commands, and not otherwise.

The effect of these three commands may be reversed by use of the corresponding *ENABLE command. Their effect otherwise persists until the user logs off.

Examples

***DISABLE**

***DISABLE SAVES**

***DISABLE LIBRARY**

Likely Errors :

There are no errors specific to this command. If the word after *DISABLE is not recognised, then the effect will be that of *DISABLE (without an argument).

Compatibility Notes :

Not supported by Acorn systems.

***DISCS**

Command Type : Transient program

Syntax : ***DISCS**

Action with Wild Cards in the File Name : Not applicable.

Description :

This program produces a list of all the discs present in the system, in order of their drive numbers. The free space on each disc is also given.

This command is identical to *FREE (q.v.), but is provided for compatibility with Acorn libraries. The system manager may delete it if it is not required.

Examples :

***DISCS**

will produce the following response from the system (for example)

Drive	Name	Bytes free	Bytes used
0	DISC1	560k	481k
1	DISC55	1098k	113k

Likely Errors :

There are no errors specific to this program.

Compatibility Notes :

Supported by Acorn systems.

***DUMP**

Command Type : Transient program

Syntax :

***DUMP <filespecifier> {<offset>}**

Action with Wild Cards in the File Name : Occurs on first match (alphabetically).

Description :

This program opens the specified file, and prints it as a hexadecimal dump on the screen of the computer. The output will start at the address <offset> (in hexadecimal) if one is specified.

The output consists of lines of the form :

aaaaaa nn nn nn nn nn nn nn ccccccc

where **aaaaaa** is the offset from the beginning of the file of the left-most byte displayed, the **nn** are the hexadecimal values of 8 bytes of the file, and the **c** are the same 8 bytes in character form. If there exists no character form (if for example the byte value is less than 32 or more than 126) then the character printed will be a dot.

This program uses the multiple byte transfer OSGBPB call, and so will run considerably faster than for example the version of *DUMP that is contained in the DFS ROM. If DFS is fitted to the computer, use ***/DUMP** to be sure of running the network version. (See under ***RUN** for full details of */).

Examples :

***DUMP FILE1**

will dump FILE1 to the BBC Microcomputer screen. A printed copy could be made at the same time by typing <Ctrl-B> before the command, to turn on the printer, and <Ctrl-C> at the end to turn it off. See section 6.5 at the beginning of this chapter about printing through the network.

***DUMP LOGFILE 1A000**

will dump LOGFILE, starting 1A000 (decimal 106496) bytes from the start of the file.

Likely Errors :

There are no errors specific to this program. However, it performs an OPENIN call, and so can cause all the same errors that the OPENIN would.

Compatibility Notes :

Supported by Acorn systems.

***ENABLE**

Command Type : File Server command

Syntax :

***ENABLE | *ENABLE SAVES | *ENABLE LIBRARY**

Action with Wild Cards in the File Name : Not applicable.

Description :

Used without a parameter, this command allows the *DELETE command to be used with wild cards, i.e. with a file specifier after *DELETE containing the characters * or #.

The system manager can set the 'Permanently ENABLEd' option in the password file for any user, to enable that user to perform wild card delete operations all the time. If this option has been set, then the *ENABLE command is redundant, unless *DISABLE has been used.

If a user attempts to use a wild card delete without either having typed *ENABLE or having the 'Permanently ENABLEd' option set, then error BD -- Not ENABLEd will be produced.

If *ENABLE is followed by SAVES, this permits the user to save files of less than 16 bytes in length (the system manager can set an option to prevent any user from saving these short files). ***ENABLE SAVES** over-rides this option, and allows any user to save short files -- its effect lasts until log-off, or until a ***DISABLE SAVES** command.

The purpose of preventing files of shorter than 16 bytes being saved is to prevent users accidentally destroying BASIC programs by saving over them after pressing <Break>, without first typing **OLD**. Users who are likely to do this are warned against the indiscriminate use of *ENABLE SAVES.

The command ***ENABLE LIBRARY** causes the currently selected library to be searched for any *, *RUN, load or 'open file for input' command. This is the default setting for SJ File Servers (but not for Acorn FSs), but it may have been turned off with ***DISABLE SAVES**

The effect of these *ENABLE commands persists until the user logs off, or until he types the corresponding *DISABLE (q.v. this Section).

Examples

***ENABLE**
***ENABLE SAVES**
***ENABLE LIBRARY**

Likely Errors :

There are no errors specific to this command. If the word after *ENABLE is not recognised, then the effect will be the same as for *ENABLE without an argument.

Compatibility Notes :

Not supported by Acorn systems.

ERAQ

Command Type : BASIC program.

Syntax :

CHAIN "ERAQ"

Action with Wild Cards : Not applicable.

Description :

This program deletes part or all of an entire directory tree.

The first question will be

Do you wish to OK the files before deletion ?

Typing Y will cause the program to prompt for Y or N after displaying the name of each file. Typing N after the first question will cause the entire directory tree to be deleted.

The program will then prompt

Full path name ?

The user should enter the full name of the directory tree (starting at the root), that he wants to delete. Entering \$ on its own would attempt to clear the entire disc (possible only for the system manager to do !). For a user to clear part or all of his files, he should enter the full path name, starting at \$.

The name of each file or directory will then be displayed, followed by the prompt for Y or N if the answer to the initial question was **Y**.

ERAQ will stop if a file or directory is found that the user does not own (i.e. have access to its account).

Note that this program will delete files even if they are locked -- users are warned to take care, especially if answering **N** to the initial question.

Examples :

CHAIN "ERAQ"

Do you wish to OK the files before deletion ? N

Full path name ? \$.JOHN.PROGS

\$.JOHN.PROGS.Main1

\$.JOHN.PROGS.Silly

\$.JOHN.PROGS.BBC.EDITOR \$.

JOHN.PROGS.BBC.eraser

\$.JOHN.PROGS

and so on, including all sub-directories contained within \$.JOHN.PROGS

Likely Errors :

There are no errors specific to this program, but ERAQ uses the *DELETE call, and so can cause the errors associated with *DELETE. There is also the possibility of getting the BASIC error **String too long** if the directory structure is very deep.

Compatibility Notes :

Supported by Acorn systems, but can be used only by a system privileged user.

*EX

Command Type :File Server command

Syntax :

***EX [<directory specifier>]**

Action with Wild Cards in the File Name : Occurs on first match (alphabetically).

Description :

This command gives information similar to that produced by *INFO, but for every file in the specified directory. If no directory is specified, then the currently selected directory will be displayed.

The information given is a header identical to that given by *CAT, followed by a line of the form below for each file :

<file name> <load addr.><execute addr.><length><access><date1><date2><time> xx(yy)

or

<directory name> No of entries=nn Default=xx/xx<access><date1><date2><time> xx(yy)

or

...Private

where

<load addr.>	is the address (in hexadecimal) in which the file would be loaded by the *LOAD or *RUN commands.
<execute addr.>	is the address (in hexadecimal) at which execution would begin in a *RUN command.
<length>	gives the true length of the file in bytes (in hexadecimal).
<access>	is the set of access letters for that item, as listed under the *ACCESS command.
<date1>	is the date that the item was originally created,
<date2>	
<time>	are the date and time that the item was last changed, i.e. written to or saved over in the case of a file, or contents changed in the case of a directory. If the date is today's, then today will appear in the <date1> and <date2> fields.
xx	is the account number associated with the file (see under *ACCOUNT)
yy	is the auxiliary account number associated with the file.
Default=xx/xx	gives the default access letters for a directory (i.e. the access status given to any file saved in it, until this is changed using the *ACCESS command). The default setting may be changed using *DEFACCESS.

The word **...Private** only is given to a non-owner of an item, if the access of the item is **P**

Examples :

***EX**

examines the contents of the currently selected directory, and will give an output similar to :

NEXTISS (027)	Owner
MAIN-IV	Option 03 (Exec)

Dir. NEXTISS	Lib. LIBRARY	
CONT	00000000 FFFFFFFF 0038A6 WR/r	11jul85 15jul85 12:55 FO (00)
GT-Example	00000800 00008023 00017C WR/r	11jul85 15jul85 14:30 FO (81)
PUTGET	00000000 FFFFFFFF 001385 WR/r	08jun85 11jun85 18:55 FO (00)
temp	00000000 FFFF3200 002231 WR/r	15jul85 15jul85 14:28 FO (00)
wombat	FFFFFFF FFFFFFFF 000000 WR/r	17jul85 today 22:32 F1 (00)

***EX \$**

examines the system root directory on the disc. This will list out all the users' root directories and any other directories and files saved in directory \$.

***EX Progs**

will examine the contents of the sub-directory Progs, which is in the currently selected directory.

***EX \$.JOHN.BBC-PROGS.OLD**

will look at directory OLD which is in BBC-PROGS in \$.JOHN

Known Bugs :

There is a bug in NFS 3.6, where the first letters *EX of a longer command beginning with *EX... will be stripped off before passing the name on to the file server. For example, ***EXAMINE** will attempt to ***RUN** a file called **AMINE**. It is wise to use ***RUN** or ***/** with any file beginning with the letters EX.

Likely Errors :

Error BE xxxx is not a directory -- if the specifier after the command is the name of a file.
 Error D6 (214) xxxx not found -- if the directory specified cannot be found.

Compatibility Notes :

Similar on Acorn systems, but without account information, or the date of original creation field. A number known as the SIN will appear in the space in which an SJ File Server would display the account numbers.

***EXEC**

Command Type: BBC Microcomputer operating system command.

Syntax:

***EXEC <file specifier>**

Action with Wild Cards in the File Name: Occurs on first match (alphabetically).

Description:

This command opens the file specified for input, and then executes text in it, as though it had been entered from the keyboard of the BBC Microcomputer. The file is closed automatically at the end of the text in it, but will remain open if an error occurs in the middle of the text sequence.

The *EXEC command is useful for performing sequences of commands repeatedly. Using either the *BUILD utility, or the *SPOOL command (q.v.), or a word processor, the sequence of commands can be built up in a file.

It is also possible to convert a text file into a BASIC program with this command. Normally, as a BASIC program is typed in at the keyboard, it is converted into a condensed token form by the language system. This is then the form in which it is saved in a file, or re-loaded. Sometimes, however, it can be more convenient to create a program using a word processing package, and then submit it to BASIC. This can be done by saving the text file, and then typing ***EXEC** followed by the file name.

Note that the *EXEC command opens the file and then reads text, using the BGET call. This is very slow on the network, so it is recommended that the user runs the utility PUTGET (q.v.) before this command, if the file to be *EXECed is very long.

Examples:

***EXEC COMMANDS**

will perform the commands contained in the file COMMANDS. If COMMANDS contains the text form of a BASIC program, this will be entered as though it were keyboard input to the currently selected language.

Likely Errors:

Error D6 (214) File not found -- if the file does not exist. Note that the system replies **File not found** (normally the File Server replies **<file name> not found**) There are no other errors specific to this command, but it calls OPENIN (q.v.) and so can cause the same errors as that command.

Compatibility Notes:

Supported by Acorn systems.

***FREE**

Command Type : Transient program

Syntax : ***FREE**

Action with Wild Cards in the File Name : Not applicable.

Description :

This program produces a list of all the discs present in the system, in order of their drive numbers. The free space on each disc is also given.

Examples :

***FREE**

will produce the following response from the system (for example)

Drive	Name	Bytes free	Bytes used
0	DISCI	560k	455k
1	Main-05	1098k	100k

Likely Errors :

There are no errors specific to this program.

Compatibility Notes :

Supported by Acorn systems.

***FSLIST**

Command Type : Transient program

Syntax : ***FSLIST**

Action with Wild Cards in the File Name : Not applicable.

Description :

*FSLIST displays a list of all active File Servers available in an installation. A File Server will not be displayed in this list if it is in Configuration or Utility Mode, or if it has crashed.

Note that, in contrast to this, *STATIONS will display File Servers, even if they are inactive.

If the Econet installation comprises multiple networks, FSLIST will also display File Servers on other networks, preceded by their network number (e.g. File Server 253 on network 2 will be displayed as **002.253**)

Examples

***FSLIST**

File Servers/Type

235	SJ Research
064:127	SJ Research
064:182	SJ Research
200	SJ Research

Likely Errors :

There are no errors specific to this command

Compatibility Notes :

Supported by Acorn systems.

***GO**

Command Type : Transient program

Syntax :

***GO <32 bit address>**

Action with Wild Cards in the File Name : Not applicable.

Description :

This command causes a jump to the specified address, which should be in hexadecimal (but with no leading & character required). If a full 32 bit address is specified, then the program looks at the most significant two bytes (four hex digits) of the address .

If the most significant two bytes of the address are FFFF, then the jump will always occur into the I/O processor (i.e. the BBC Micro itself), to the address given by the least significant two bytes. For example ***GO FFFF2084** will jump to address 2084 (hex) in the BBC Micro.

If the most significant two bytes of the address have any value other than FFFF, then this command will jump into the second processor, if one is present, otherwise into the I/O processor.

There is a *GO command built in to the second processor 'tube' interface, but this does not force a jump to the I/O processor if the more significant half is FFFF. If this feature is required, you must type */GO

Obviously it is vital that you know what you are doing before executing this command, otherwise the most likely outcome is that the computer will crash.

Examples

***GO 3000**

will jump to address 3000 (hex) in the second processor (if any), or in the BBC Micro if no second processor:

***GO FFFF3000**

will invariably jump to address 3000 (hex) in the BBC Microcomputer itself.

Likely Errors :

If the incorrect syntax is given to *GO, the message **Syntax: *GO <32 bit address>** will be displayed, and error number DC (220) will be given.

Compatibility Notes

Supported by Acorn systems.

***GTIME**

Command Type : Transient utility program

Syntax :

***GTIME**

followed immediately by 5 GET statements

Action with Wild Cards in the File Name : Not applicable.

Description :

This command plants 5 characters into the BBC Microcomputer, so that they can be read by a program using five GET commands (which must follow *GTIME immediately).

After the two lines above have been run, the values read will be (in order) :

first byte = day of month (between 1 and 31)
second byte AND &OF = month (between 1 and 12)
(second byte AND &FO) DIV 16 = years after 1981 (e.g. 1985 will read as 4)
third byte = hours from midnight (between 0 and 23)
fourth byte = minutes (between 0 and 59)
fifth byte = seconds (between 0 and 59)

Do not use BASIC keyword INPUT with *GTIME, since any bytes which are interpreted as 'control' characters will not be input, but will be interpreted directly by the input routine. Also it is wise to ensure that the piece of program containing the GET statements is absolutely correct, since if BASIC finds an error here, the 5 'time' bytes will be entered into the keyboard buffer, possibly causing strange effects.

Examples :

```
DIM t% 4  
*GTIME  
day%=GET  
monthandyear %= GET  
hours%=GET  
mins%=GET  
secs%=GET
```

```
IF hours%<12 THEN am$="in the morning"  
IF hours%=0 THEN am$="midnight"  
IF hours%=12 THEN am$="noon"  
IF hours%>12 THEN am$="in the afternoon"  
IF hours%>17 THEN am$="in the evening"  
IF hours%>12 THEN hours%=hours%-12  
IF hours%=0 THEN hours%=12
```

```
PRINT "It is now ";mins% " minutes past ";hours%;" "am$
```

Likely Errors :

There are no errors specific to this command, but the BBC Microcomputer screen may do strange things if a mistake occurs between executing *GTIME and GETting 5 bytes.

Compatibility Notes :

Supported by Acorn systems that contain a real-time clock (otherwise the time may be nonsense).

***GUSER**

Command Type : Transient utility program

Syntax :

```
*GUSER  
INPUT user$
```

Action with Wild Cards in the File Name : Not applicable.

Description :

This program plants the name of the user logged on from this computer, into the computer so that it can be read into a string by the BASIC keyword INPUT -- this must follow *GUSER immediately.

User names have a maximum length of 10 characters.

If a BASIC error occurs between *GUSER and the INPUT statement, then the user name will be entered into the keyboard buffer, and will probably cause BASIC to reply **Mistake**

Examples

```
*GUSER  
INPUT U$  
PRINT "Hallo "U$ ", how are you today ?"
```

Likely Errors :

There are no errors specific to this command

Compatibility Notes :

Supported by Acorn systems, but note that some Acorn File Servers support user names of more than 10 characters.

***I AM**

Command Type : N.F.S. command

Syntax :

***I AM [<file server number>] <user id> [<password>]**

***I AM [<file server number>] <user id> :
[<password>]**

Action with Wild Cards : Allowed in user id field. Will give first match (alphabetically).

Description :

This command 'logs-on' a user to a File Server -- that is to say, it identifies the user to the file server, and sets up communication channels between the user's computer and the File Server.

The *I AM command must be the first command given to the file server, otherwise it will reply with the error message **Who are you?** or **Channel** when any filing operation is attempted through the network.

The file server will search the password file on each disc in turn (starting with the lowest numbered drive) for the <user id>, and will check the password quoted by the user against any that he may have set up using the *PASS command previously. The file server will read from the password file the list of accounts to which the user has access, whether the user has system privilege, and will select a library directory (usually \$.LIBRARY, unless otherwise set up by the system manager) for the user. The system will also search the disc on which the user's password file entry was found, for a directory with the same name as the user, and will select this as his User Root Directory (URD) -- see Section 6.2.

If the <user id> is not found in the password file of any disc, the user will not be able to log on unless the system manager has set up a default user, in which case an attempt to log-on with a user identifier unknown to the system will leave the user logged on as this default user. The system manager will normally have set up some automatic response for the default user, for example to prompt the user to log on again.

The BBC Microcomputer initially assumes that the file server station number is 254, unless this is otherwise specified. If the required file server is at another station (note that there may be more than one File Server on Econet), then type (for example) .

***I AM 250 FRED**

to log-on to a file server at station 250. Any subsequent *I AM command will now assume file server 250, until another *I AM command specifies a different file server station number.

If the network contains bridges on to other networks, then it is possible to log on to File Servers on these other networks. To do so, the user must specify the full station number of the File Server, which will be of the form **<network number>.<station number>** For example, to log on to the File Server at station 254 on network number 3, type

***I AM 3.254 FRED**

The user does not need to re-specify the network number for further operations until he wants to select a different network. For example, to log on to the local File Server station 254, (the local network is always number 0) type :

***I AM 0.254 FRED**

The user may conceal his password from prying eyes by typing a space then a colon after his user identifier. Follow this by <return>, then type in the password, which will then not appear on the screen. Note that this feature works only with BBC Microcomputers equipped with NFS 3.6 or later.

The utility program *LOGON (q.v. this Section) has a similar effect, but works for all versions of NFS. The use of *LOGON is generally recommended for all users.

If a *I AM command is typed at a particular station, and the File Server finds that someone is already logged on at that station number, then the old user will automatically be logged off, and the new one logged on. Note that users are not logged off under any other circumstances, nor are they logged off from one File Server if they log on to a different one. The use of the command *BYE at the end of a session is recommended, and especially vital if security is important.

Examples :

***I AM JOHN**

logs on to the previously specified File Server (or the File Server at station 254 if none previously specified) with user identifier JOHN:

***I AM 235 SMITH**

attempts to log-on user SMITH to the File Server at station 235.

***I AM SYST :**

Type <return> after the colon, then your password. The password will be concealed from view. (Note that this works only with NFS 3.6 or later in the BBC Micro)

Likely Errors :

Error A2 (162) Not listening -- Either there exists no File Server at the specified station number, or it has been switched off. Press <Ctrl-Break> on your computer and try again. If this fails, then ask your system manager to help.

Error A3 (163) No clock -- The user's computer is not plugged into the network, or the network clock is not running. Ask your system manager.

Error B9 (185) Bad password -- An attempt to use illegal characters in the password field (e.g. * # \$ %) will give this error.

Error BB (187) Wrong password -- if the user makes an error in entering his password.

Error BC (188) User not known -- If there is no default user, this error will be given if the user's name is not known to the system. Ask your system manager.

Error D6 (214) File not found -- if the file !BOOT does not exist in either the URD or library. Note that the system replies **File not found** (normally the File Server replies <file name> **not found**). You are however logged on at this stage despite this error message.

Compatibility Notes :

Supported on Acorn systems

*INFO

Command Type :File Server command

Syntax :

***INFO [<general specifier>]**

Action with Wild Cards in the File Name : Occurs on first match (alphabetically).

Description :

This command prints full information about the file (or directory) specified, or the currently selected directory if none specified. The displayed information for the item is :

<file name> <load addr.><execute addr.><length><access><date1><date2><time> xx(yy)

or

<directory name> No of entries=nn Default=xx/xx<access><date1><date2><time> xx(yy)

or **...Private**

where

<load addr.> is the address (in hexadecimal) at which the file would be loaded by the *LOAD or *RUN commands

<execute addr.> is the address (in hexadecimal) at which execution would begin in a *RUN command

<length> gives the true length of the file in bytes (in hexadecimal)

<access> is the set of access letters for that item, as listed under the *ACCESS command.

<date1> is the date that the item was originally created,

<date2>

<time> are the date and time that the item was last changed, i.e. written to or saved over in the case of a file, or contents changed in the case of a directory. If the date is today's, then **today** will appear in the <date1> and <date2> fields.

xx is the account number associated with the file (see under *ACCOUNT)

yy is the auxiliary account number associated with the file.

Default=xx/xx gives the default access letters for a directory (i.e. the access status given to any file saved in it, until this is changed using the *ACCESS command). The default setting may be changed using *DEFACCESS.

The word **...Private** only is given to a non-owner of an item, if the access of the item is P

If this listing is required for all files in a directory, use the *EX command (q.v. this Section).

Examples :

***INFO PROGRAMS**

will display a line of information as above for the file (or directory) PROGRAMS.

***INFO ***

will give information on the first entry (only) in the currently selected directory.

***INFO \$.F RED**

will give information on \$.FRED, which is probably a user root directory for user FRED.

***INFO**

gives information on the currently selected directory.

Likely Errors :

Error D6 (214) Not found -- if the specified item is not found.

Compatibility Notes :

Supported by Acorn systems.

***LIB**

Command Type : File Server command

Syntax :

*LIB <directory specifier>

Action with Wild Cards in the File Name : Occurs on first match (alphabetically).

Description :

This command selects the specified directory as the current library directory.

When any **CHAIN LOAD OPENIN OPENUP *EXEC *LOAD *RUN** or *<file specifier> command is given, the user's currently selected directory will be searched for the file. If it is not found, the currently selected library will be searched, and the file opened or loaded from there if found.

The OSFILE machine code call with A=&05 (ROI) and A=&FF (load) (See Econet Advanced User Guide page 38) will also check the library directory if the file specified is not found in the currently selected directory.

Note that the *I AM command (q.v.) will automatically select as library directory \$.LIBRARY on the lowest numbered disc drive if such a directory exists, hence this command is needed only if a different directory is to be specified as the library.

See the *DISABLE command in this Section, which makes the library searching equivalent to that in an Acorn File Server (i.e. searched for * and ***RUN** commands only).

Examples :

***LIB \$.OTHERLIB**

to select this alternative directory as the library.

Likely Errors :

Error BE (190) xxxx is not a directory -- if a file name has been specified.

Error D6 (214) xxxx not found -- if the directory specified does not exist.

Compatibility Notes :

Supported on Acorn systems. Note that the library search is done only for *<file specifier> (or ***RUN**) commands, and not for any of the other operations specified above. Acorn systems will automatically select as library a directory called \$.LIBRARY on the same disc as the user root directory is found : this is slightly different from the rules on a SJ Research system, where the directory specified in EDITPASS (usually \$.LIBRARY on the lowest numbered drive) is selected.

LOAD

Command Type : BASIC command

Syntax :

LOAD "<file specifier>" | LOAD <string variable>

where the <string variable> must contain a legal file specifier.

Action with Wild Cards in File Name : Occurs on first match (alphabetically).

Description :

This BASIC command causes the file with name equal to the string immediately following the LOAD command to be copied into memory as though it were a BASIC program. An error message will be generated by the BASIC language system if the file specified did not contain a BASIC program.

The action taken by the file server is to search through the currently selected directory for the file specified. If it is not found, the file server will search through the currently selected library (see under *LIB command), and will load the file from there if found.

Examples :

LOAD "COPIER"

loads this file from the currently selected directory (CSD), or from the currently selected library if it was not found in the CSD. Note that any string constant in BASIC must be enclosed in quotes.

LOAD FILE\$

loads the file whose name is equal to the string FILE\$. Again the currently selected library is searched if the file is not found in the CSD.

Likely Errors :

Error B5 (181) xxxx is a directory -- you cannot LOAD a directory.

Error BD (189) Insufficient access -- There must be access status **R** for this user, otherwise he will not be able to load the file.

Error CC (204) Bad name -- if the file name contains illegal characters (e.g. \$ % . ^ : except in contexts where they are permitted)

Error D6 (214) xxxx not found -- if the specified file did not exist.

Compatibility Notes :

Supported by Acorn systems. Note that Acorn File Servers do not perform a library search with the LOAD command.

***LOAD**

Command Type : BBC Microcomputer operating system command

Syntax :

***LOAD <file specifier> [<load address>]**

Action with Wild Cards in File Name : Occurs on first match (alphabetically).

Description :

This command causes the contents of the specified file to be copied into memory. *LOAD is followed by the file specifier, and may optionally take a second parameter, the base address (in hexadecimal) for the copy in memory. If this base address is not specified, the base address will be equal to that in the *SAVE command (q.v.).

The action taken by the file server is to search through the currently selected directory for the file specified. If it is not found, the file server will search through the currently selected library (see under *LIB command for details), and will load the file from there if found.

Examples :

***LOAD DATA**

loads this file from the currently selected directory (CSD), or from the currently selected library if it was not found in the CSD. The contents of the file will be loaded at the address recorded by the *SAVE command when the file was saved.

***LOAD "FILE*" FFFF7C00**

loads the first file (alphabetically) matching the wild card specifier. Again the currently selected library is searched if the file is not found in the CSD. The file will be loaded into the BBC Microcomputer (not any second processor) at address 7C00, which is the base of the Mode 7 screen.

Likely Errors :

Error B5 (181) xxxx is a directory -- you cannot *LOAD a directory.

Error BD (189) Insufficient access -- There must be access status **R** for this user, otherwise he will not be able to load the file.

Error CC (204) Bad name -- if the file name contains illegal characters (e.g. \$ % . : except in contexts where they are permitted)

Error D6 (214) xxxx not found -- if the specified file did not exist.

Compatibility Notes :

Supported by Acorn systems. Note that Acorn File Servers do not perform a library search with the *LOAD command.

***LOGON**

Command Type : Transient program

Syntax : ***LOGON**

Action with Wild Cards : Not applicable.

Description :

This program prompts a user for user identifier and password, then logs that use on to a File Server. It is recommended that *LOGON is used (in a boot file as described below) wherever security is important.

The password is not reflected on the computer screen, but an asterisk appears for each keystroke.

This program can be put into file \$.BOOT.!BOOT (and user BOOT should have *OPT4,3 set), so that any user can press <Shift-break> to run it automatically. See this Section under *OPT4 for further details. This relies on the File Server being station 254 -- this is the default assumed by the BBC Microcomputer.

*LOGON is particularly useful with BBC Microcomputers equipped with NFS 3.34, as this does not support the ***I AM <user id.>** : option to conceal passwords (See *I AM command). It also sets the protection byte, so that other users cannot examine the memory of a computer while it is logging on, and erases any copy of the password from the computer memory after log-on.

Examples :

***LOGON**

User : FRED

Password : *****

Likely Errors :

This program calls ***I AM**, and so can cause any of the errors that *I AM could cause.

Compatibility Notes :

Supported by Acorn systems.

MULTICOPY

Command Type : BASIC program

Syntax :

CHAIN "MULTICOPY"

Action with Wild Cards in directory names : Occurs on first match (alphabetically).

Description :

This program copies entire directory trees between File Servers, or between different places in the same File Server.

It will prompt for the log-on text for the File Server containing the source files, and the same for the destination File Server.

The program will then ask **Do you wish to include sub-directories ?** -- if the user answers **Y**, it will copy the entire set of sub-directories and the files in them. It will also ask if the account information is to be copied -- if the answer to this question is **N**, the accounts of all the files and directories copied will be that of the directory into which they are being copied.

The user must own the destination directory, and have read access to all the files to be copied.

Examples :

CHAIN "MULTICOPY"

Multiple file copy utility V0.10

MULTICOPY copies groups of files from one file server to another. It may also be used between directories or discs on the same file server.

Log-on text for source FS:

***I AM 254 FRED**

Log-on text for dest. FS

(or press RETURN for same FS) :

***I AM 253 FRED**

Do you wish to include sub-directories (YIN) : Y

Do you wish to copy account information (Y/N) : Y

source directory name : PROGS

destination directory name : PROGS

·
·
·

(list of the files being copied)

Likely Errors :

Error BD (189) Insufficient access -- if the user does not have access **R** to all the files to be copied from the source, or does not own the destination directory.

Error BE (190) xxxx is not a directory -- if the user has specified a file as the source or destination directory name prompt.

Error C2 (194) Already opened by xxxx -- MULTICOPY will save over a file of the same name. If this file was already open, this error will occur.

Error C3 (195) Locked -- after an attempt to save over a file of the same name, if the latter was locked.

Error D6 (214) xxxx not found -- if the source or destination directories could not be found.

Compatibility Notes :

Supported by Acorn systems, except that accounts do not exist, and so an attempt to copy account information across will cause an error. Since Acorn systems use the root of user's tree of directories to determine its ownership (rather than account numbers), a user will not have owner access to files specified as **\$.<directory name>.<file name>**. To get round this problem, it is wise to log on as a system privileged user.

***NOTIFY**

Command Type : Transient program.

Syntax :

***NOTIFY <station number> <text> | *NOTIFY <user id.> <text>**

Action with Wild Cards in User Identifier : Occurs on first match, from the top of the list as produced by *USERS (q.v.) downward.

Description :

This utility produces the message <text> on the screen of the specified station, or of the station where a specified user is logged-on.

If the specified user is logged-on at several stations, the message will appear only at one of them : this will be the station at which the specified user last performed a filing operation, and also the first one that would appear in the list produced by the program *USERS (q.v.).

The station specified (or found by the above process from the user identifier) must be switched on and connected to the network, otherwise the error message **Not listening** will be given. If the user has run ***PROT** or otherwise set the protection byte in his computer, the **Not listening** message will also be produced.

Examples :

***NOTIFY FRED HELLO THERE**

will send the message HELLO THERE to the station at which user FRED last performed a network filing operation.

***NOTIFY 3 TIME FOR LUNCH**

will send the message to station 3, if it is switched on. The message produced in each case will be of the form (if the message had been sent from station 5)

***| 005: TIME FOR LUNCH**

Likely Errors :

Error A1 (161) Net error -- if the line is more than 80 characters in length. This is due to overflow of the File Server command line buffer. Use several *NOTIFY commands for long messages, or use a program to call OSWORD with A=14 (see §8).

Error A2 (162) Not listening -- if the destination station is switched off, or if its protection byte has been set.

Error AE (174) Not logged on -- if a user identifier was specified, and that user was not logged on at the currently selected File Server.

Compatibility Notes :

Supported on Acorn systems, but note that the user table is not re-ordered to reflect the latest operation, but is effectively in random order, so that *NOTIFY <user id.> will not necessarily find that user.

OPENIN

Command Type : BASIC 2 keyword

Syntax :

<numeric variable>=OPENIN "<file specifier>"
<numeric variable>=OPENIN <string variable>

Action with Wild Cards in File Name : Occurs on first match (alphabetically).

Description :

This BASIC command opens a file for random access using BGET#, INPUT# commands. OPENIN is followed by a file specifier, and is an integer function, returning a value called the channel for the file : if the channel is zero, this means that the file was not found, and no other error will be given. OPENIN opens a file for reading only. an attempt to write to the file will cause an error. The user must have access **R** to the file to be able to open it for input.

The action taken by the file server is to search through the currently selected directory for the file specified. If it is not found, the file server will search through the currently selected library (see under *LIB command for details), and will open the file from there if found. If a library search is not wanted, then the programmer should specify either the full file specifier (beginning with \$), or use 9 to specify the currently selected directory. See also *DISABLE LIBRARY command in this Section, to restrict the library search.

A file may be opened for input even if other users have already opened it for input also. However, it is not permitted to open a file for input if it already open for output or update.

It is also permissible to open a directory using this command, but it is not possible to read any data from it: an attempt to do so will cause an error. This keyword could be useful to check for the existence of a directory. Note that OPENIN will not search the library for a directory (but it will, as normal, for a file).

There is a limit to the number of files that a user may have open at once. This is initially 8, but the user root directory (URD) and currently selected library are opened by the system at log-on, leaving a maximum of 6 for the user. In addition, if the user specifies a currently selected directory other than the URD or library, this is also opened; also the *DIR command uses a further channel fleetingly. Hence it is wise for programmers to rely only on having four channels for random access filing operations.

The OPENIN command in BASIC 1 is in fact OPENUP. This means that you cannot use it to open a file to which you have access **R** only, if your BBC Microcomputer is equipped with BASIC 1. To find which version is fitted, type *BASIC followed immediately by **REPORT**. The response will be either **1981** for BASIC 1, or **1982** for BASIC 2. To open a file for reading only, use the function FNOpenin given below:

Note also that the command OPENIN in BASIC 1 has the same internal representation as the command OPENUP in BASIC 2. This means that a file initially run and saved in BASIC 1 will run identically on BASIC 2 -- furthermore, when the program is listed in BASIC 2, the translation to OPENUP will have happened automatically.

However, a BASIC 2 program containing OPENIN commands will not run in BASIC 1, and these commands will not appear in a BASIC 1 listing -- hence if compatibility is required between BASIC 1 and BASIC 2 for opening read-only files, the function FNOpenin should always be used.

Examples :

A%=OPENIN "DATAFILE"

followed later in the program by

INPUT#A%, X

reads the value of X from file DATAFILE. The system will look in the currently selected library for the file, and open that, if the file is not present in the currently selected directory.

```
channel%= OPENIN "$ JOHN.PROJECT.DATA"
```

sets up to read from a file in sub-directory PROJECT of directory \$.JOHN (and will not search the library if file DATA is not found). Similarly,

```
channel%=OPENIN"@.data23"
```

will look for **data23** only in the currently selected directory, and not in the library.

Program FNOpenin for use with BASIC 1 :

The BASIC 1 language does not support OPENIN as described here. It contains a keyword OPENIN, but it is in fact OPENUP (q.v.). The function FNOpenin listed below is recommended if you need to open a file for reading only (in particular, when the user has only access R to the file)

At the head of the program,

```
DIM combuf 128  
osfind=&FFCE
```

and after the body of the program,

```
DEFFNOpenin($combuf)  
LOCAL A%, X%, Y%  
A%=&40  
X%=combuf  
Y%=X% DIV 256  
=(USR osfind) AND &FF
```

then, in the body of the program, use (for example)

```
channel%=FNOpenin("Data")
```

then check that channel% is non-zero, and **byte=**

```
BGET# channel%
```

Likely Errors :

Error BD (189) Insufficient access -- There must be access status R for this user, otherwise he will not be able to open the file for input. If the BBC Microcomputer is equipped with BASIC 1, it will be necessary to use the program given above.

Error C0 (192) Too many files open -- There is a limit to the number of channels available, normally 5 or 6.

Error C1 (193) File not open for update -- will be caused by an attempt to use BPUT# or PRINT# after OPENIN.

Error C2 (194) Already opened by xxxx -- It is not permitted to open a file for input, if it already open for output by either this user or another.

Error CC (204) Bad name -- if the file name contains illegal characters (e.g. \$ % . : except in contexts where they are permitted)

Compatibility Notes :

Supported on Acorn systems. Note that Acorn File Servers do not perform a library search with the OPENIN command.

OPENUP

Command Type : BASIC 2 keyword

Syntax :

```
<numeric variable>=OPENUP "<file specifier>" |  
<numeric variable>=OPENUP <string variable>
```

Action with Wild Cards in File Name : Occurs on first match (alphabetically).

Description :

This BASIC command opens a file for random access using BGET#, BPUT#, INPUT#, PRINT# commands. OPENUP is followed by a file specifier, and is an integer function, returning a value called the channel for the file : if the channel is zero, this means that the file was not found, and no other error will be given. OPENUP opens the file for reading or writing.

The user must have at least access **W** to the file to use this command, and would normally have access **R** as well, so that he could read from or write to the file. There is however a meaning to write-only access: this is append only -- the file may only be written to if the pointer PTR# is equal to the extent of the file EXT#.

The action taken by the file server is to search through the currently selected directory for the file specified. If it is not found, the file server will search through the currently selected library (see under *LIB command for details), and will open the file there if found. To inhibit the library search, either specify the full file name (beginning with \$), or use @,<file name>. See also *DISABLE LIBRARY command in this Section, to restrict the library search.

A file may be not opened for update if it has already been opened for input or update by this or another user.

There is a limit to the number of files that a user may have open at once. This is initially 8, but the user root directory (URD) and currently selected library are opened by the system at log-on, leaving a maximum of 6 for the user. In addition, if the user specifies a currently selected directory other than the URD or library, this is also opened; also the *DIR command uses a further channel fleetingly. Hence it is wise for programmers to rely only on having four channels for random access filing operations.

This keyword does not exist in BASIC 1, but the BASIC 1 keyword OPENIN has exactly the same effect. To find which version is fitted, type *BASIC, followed immediately by **REPORT**. The response will be either **1981** for BASIC 1, or **1982** for BASIC 2.

Note also that a program run in BASIC 2, containing OPENUP commands, will run satisfactorily on BASIC 1 -- when it is listed in BASIC 1 the commands will have changed into the BASIC 1 keyword OPENIN (q.v. this Section).

Examples :

```
A%=OPENUP "DATAFILE"
```

followed later in the program by a check that A% is non-zero, then

```
PRINT#A%, X
```

writes the value of X to file DATAFILE. The system will look in the currently selected library for the file, and open that, if the file is not present in the currently selected directory.

```
channel%=OPENUP"$JOHN.PROJECT.DATA"
```

sets up to read from a file in sub-directory PROJECT of directory \$.JOHN

Likely Errors :

Error B5 (181) xxxx is a directory -- you cannot OPENUP a directory.

Error BD (189) Insufficient access -- There must be access status **W** for this user, otherwise he will not be able to open the file for for updating.

Error C0 (192) Too many files open -- There is a limit to the number of channels available, normally 5 or 6.

Error C1 (193) File not open for update -- caused after OPENUP by a user only having access **W** to a file attempting to read the file, or to write to the file without PTR# being equal to EXT#.

Error C2 (194) Already opened by xxxx -- It is not permitted to open a file for update, if it already open for any purpose by either this user or another.

Error CC (204) Bad name -- if the file name contains illegal characters (e.g. \$ % . ^ : except in contexts where they are permitted)

Compatibility Notes :

Supported on Acorn systems. Note that Acorn File Servers do not perform a library search with the OPENUP command, nor do they support the 'append only' feature resulting from **W** only access to a file.

OPENOUT

Command Type : BASIC keyword

Syntax :

<numeric variable>=OPENOUT "<file specifier>" |
<numeric variable>=OPENOUT <string variable>

Action with Wild Cards in File Name : Wild cards prohibited

Description :

This BASIC command creates a new file with name <file specifier>. It will delete any existing file of the same name. The file is opened for reading or writing, although it will be necessary to write some data first, otherwise an EOF error will occur. It will be possible to write to the file using BPUT# or PRINT# even if the default access for the directory containing the file is R only.

It is an function, returning a value between 0 and 255. If this number is non-zero, it is called the channel for the file. A value of zero means that a directory specified was not found -- zero is not returned for any other reason (and can therefore be used to check for the presence of a directory).

The length initially assumed for the file is zero: file space will be allocated according to the value of PTR#. The minimum non-zero file size is 1 kilobyte, and the length will be increased in kilobyte steps when PTR# becomes more than the file size. Note that the file server will allocate 1 Kbyte pages only when they are required (i.e. when they are written to). Hence it is possible to have a file with extent of 10 Kbytes, but only using 2 Kbytes of disc space, because there is data only near the beginning and end.

There is a limit to the number of files that a user may have open at once. This is initially 8, but the user root directory (URD) and currently selected library are opened by the system at log-on, leaving a maximum of 6 for the user. In addition, if the user specifies a currently selected directory other than the URD or library, this is also opened; also the *DIR command uses a further channel fleetingly. Hence it is wise for programmers to rely only on having four channels for random access filing operations.

Examples :

A%=OPENOUT"NEWDATA"

followed later in the program by

BPUT#A%, CH

which will write the least significant single byte of CH to the file NEWDATA

Likely Errors :

Error B5 (181) xxxx is a directory -- caused by attempting to create a file with the same name as that of a directory.

Error BD (189) Insufficient access -- caused if the user is not an owner (i.e. does not have access to the main or auxiliary account) of the directory in which the new file is to be created.

Error C0 (192) Too many files open -- There is a limit to the number of channels available, normally 5 or 6.

Error C2 (194) Already opened by xxxx If this or another user has this file open for reading or writing, then it cannot be deleted by creating a file with the same name, until it has been closed.

Error C3 (195) Entry locked -- It will be necessary to use the *ACCESS command to unlock the file before it can be deleted by creating a file with the same name.

Error CC (204) Bad name -- if the file name contains illegal characters (e.g. \$ % . ^ : except in contexts where they are permitted)

Compatibility Notes :

Supported by Acorn systems, but the space allocation works differently.

***OPT1**

Command Type : BBC Microcomputer operating system command

Syntax : ***OPT1,<number>**

Action with Wild Cards in File Name : Not applicable.

Description :

This command controls the display of information after any filing operation. The value of the number can be 0 or 1, and has the following effect:

***OPT1,0** No information displayed

***OPT1,1** The following information is displayed after any **LOAD SAVE *LOAD *SAVE *RUN** or ***<file specifier>** commands only.

<file name> <load address> <execute address> <length>

The addresses and length are as displayed in *INFO or *EX, and are in hexadecimal. Specifying a <number> in excess of 1 has the same effect as *OPT1,1. (Note that this is true only of the Econet system, and not necessarily of other filing systems for the BBC Microcomputer).

The setting of OPT1 remains only for the duration of the session, although it is preserved over <Break>. The setting is lost if the power is turned off to the BBC Microcomputer, or if <Ctrl-Break> is pressed.

Examples :

***OPT1,1**

sets the flag to cause additional information to be printed after certain filing operations (as above).

Known Bug :

In NFS 3.6, if OPT1,1 is set, and printing through the network is in progress (i.e. character <Ctrl-B> has been output), then it is possible that a LOAD or SAVE operation that fails will not generate an error message to the user (in fact this occurs when the network printer buffer is nearly full). Care should be taken with the use of OPT1,1 if printing is likely.

Likely Errors :

Error FE (254) Bad command -- if the second number is in excess of 255.

Compatibility Notes :

Supported on Acorn systems. Some versions will not work with the value of <number> greater than 1.

*OPT4

Command Type : BBC Microcomputer operating system command

Syntax : *OPT4,<number>

Action with Wild Cards in File Name : Not applicable.

Description :

This command controls the automatic execution of commands when a user logs on using the *I AM command. The number after *OPT4 can be between 0 and 3, and has the following effect:

- *OPT4,0 specifies no action at log-on
- *OPT4,1 performs *LOAD !BOOT i.e. the file is loaded into memory, but no further action is taken
- *OPT4,2 performs *RUN !BOOT i.e. runs !BOOT as though it were a machine code program
- *OPT4,3 performs *EXEC !BOOT, i.e. it obeys commands in !BOOT as though they had been typed from the keyboard. This is probably the most useful option, as it allows file server and other commands to be put easily into the boot file.

If *OPT4,x has been set at some time with x not zero, the file server will search for the file !BOOT at log-on. The user root directory will be searched first, and then the library directory. If !BOOT is not found, the message **File not found** will be displayed: note that this does not mean that log-on has failed, but only that !BOOT was absent. (Note also that SJ File Servers do not usually give **File not found** -- under most other circumstances <file name> **not found** will appear).

If the only file !BOOT in the system is in the library, then it would be possible to use it to produce a message of the day at log-on, which would be displayed every time a user (who has *OPT4,3 set) logged-on.

The system manager can, for each user, lock the boot option so that the user cannot change it -- an attempt to do so will cause error C3 (see below).

Examples :

*OPT4,2

sets the auto-run option so that file !BOOT is loaded and run as a machine code program at log-on.

Likely Errors :

Error BA (186) Insufficient privilege -- There is an option available for the system manager to 'lock' the password and *OPT4 setting, so that a user cannot change them. If this option has been set, this error will be given.

Error 03 (3) Password file changed -- caused if the system manager has made any change to the password file since the user logged on. The user should log on again.

Compatibility Notes :

Supported on Acorn systems, but note that the library search is performed only after *OPT4,2 is set with Acorn File Servers.

OSCLI

Command Type : BASIC 2 keyword.

Syntax :

OSCLI "<string>" | OSCLI <string variable>

Description :

This command is not associated specifically with the Econet system, but this description is supplied because it does not appear in any other official documentation.

OSCLI is implemented only on BBC BASIC 2, and not on BASIC 1. Its effect is to pass the string specifier direct to the operating system Command Line Interpreter (CLI), so that **OSCLI "<string>"** has the same effect as ***<string>**.

Unlike * commands in BASIC, OSCLI can be used within a multi-statement line, and with string variables. This means that programs can accept * commands in place of their normal input, and then execute them.

For programs that need to run on BASIC 1, the procedure PROCOScli defined below should be used.

Procedure for use with BASIC 1 :

DIM buf% 128

within the main body of the program, followed by, after the end,

```
DEF PROCOScli($buf%)  
LOCAL X%, Y%  
X%=buf%  
Y%=X% DIV 256  
CALL &FFF7  
ENDPROC
```

The procedure could be called by (for example) :

```
270 INPUT source$  
280 IF LEFT$(source$)="*" THEN PROCOScli(source$): GOTO 270
```

Likely Errors :

There are no errors specific to this command, but any error generated by the command sent to the CLI will occur as normal.

Compatibility Notes :

Supported on Acorn systems.

***PASS**

Command Type : File Server command

Syntax :

***PASS <old password> <new password>**

Action with Wild Cards in File Name : Wild cards prohibited.

Description :

This command changes the user's password. The system manager may have set a password up for you, in which case you must quote the old password followed by the new one. If no password had been set then it is necessary to quote a null string "" as the old password.

Passwords have a maximum length of 10 characters, and may contain all characters permitted in file names, i.e. alphanumeric and ! - Upper and lower case letters are treated as equivalent.

The system manager can set an option to prevent you from changing your password. If this has been done, you will get error C3 if you attempt to change it.

If security is important to you, then take care in your choice of a password. We recommend against using any character string that others might guess (e.g. your wife's name, your phone number, car number, etc!). It is probably best not to use a normal word, since someone may see part of it and then guess the rest. Finally, use as many characters as possible (it does not take long to run a program to try all 4-character passwords).

Examples :

To set up your password as HELLO type

***PASS "" HELLO**

(assuming no password set previously). Conversely, to clear the password, type

***PASS HELLO ""**

Likely Errors :

Error 03 (3) Password file changed -- this error will be produced if the password file has been changed by the system manager, while the user is logged on. The user -- should log on again.

Error B9 (185) Bad password -- There is an illegal character in the password quoted, probably * # \$ % ^ :

Error C3 (195) Entry locked -- caused if the system manager has set the 'lock password' option for this user.

Error BB (187) Wrong password -- The old password does not match the one stored.

Compatibility Notes :

Supported by Acorn systems. Note that the option to 'lock' the password does not exist on Acorn File Servers.

***PRINT**

Command Type : Transient program.

Syntax :

***PRINT**

Action with Wild Cards : Not applicable.

Description :

The Econet software in the BBC Microcomputer assumes that the network printer server has number 235 unless specified otherwise by this command. The SJ Research File Server contains a printer server, so if the File Server is station 254 and the built-in printer server is used, the *PRINT (or *PS command below) command will be necessary to get any output.

*PRINT is now superseded by the command *PS below, which also performs automatic selection of printer servers if there is more than one in a network, and allows the user to specify printers by name.

*PRINT combines the effects of ***FX 5,4** (select network printer) ***FX 6,0** (allow line-feeds through) and ***PS <file server station number>** described above. It thus performs all the commands usually required to select the network printer.

Users who have automatic line feed selected on their printers can modify this program to remove the ***FX6,0**. To do so, log on as a system privileged user, then type :

```
*DIR $.LIBRARY  
H%=OPENUP "PRINT"  
PTR#H% = &A  
BPUT#H%,10  
CLOSE#H%
```

Likely Errors :

There are no File Server errors specific to this program.

It is possible that any program calling *FX5 may not work if the local printer buffer is not empty -- if, for example, a local printer has been used but had not completed the job. If *PRINT appears to hang up, press <escape> <Ctrl-B> <escape>, then try *PRINT again.

Compatibility Notes :

Supported by Acorn systems. Note that *PRINT only works where the File Server is also the printer server -- but it is recommended that they are combined in Level 2 systems.

***PRINTER**

Command Type : File Server command

Syntax :

***PRINTER | *PRINTER <name>**

Action with Wild Cards in the File Name : Not applicable.

Description :

This command is intended for use with the *PRINTOUT command (q.v. this Section). The user must be logged on to a File Server, and the *PRINTER command will apply to this File Server.

Used without an argument, this command displays the name and status of the currently selected printer. The possible statuses are :

ready	the printer is available for use
busy	the printer will be available for use when the current work is finished
offline	the printer is not connected, or has run out of paper, ribbon etc.
not authorised	this user is not authorised to use the selected printer.

If *PRINTER is followed by a printer name, then the internal printer selection within the current File Server, will be changed to the new name. Error 10 will be produced if the user is not authorised to select the printer specified. See Section 6.5 for a description of printer names.

*PRINTER will not change the printer server station number in the BBC Microcomputer, but if the printer server station number has been selected to be the same as the File Server, it will change the File Server's internal selection, and therefore will change which printer is used for network printing.

Examples

***PRINTER**

will produce, for example

EPSON ready

***PRINTER ML**

which will produce a response, for example

ML busy with station 003 (FRED)

Likely Errors :

Error 10 (16) Not authorised to use printer -- if the system manager has restricted the use of this printer to users with access to a certain account.

Compatibility Notes :

Not supported by Acorn systems.

***PRINTOUT**

Command Type: File Server command

Syntax:

***PRINTOUT <file specifier>**

Action with Wild Cards in File Name: Occurs on first match only (alphabetically)

Description:

This command causes the specified file to be printed, at the currently selected printer as selected by *PS (i.e. the same printer that would be used if normal printing were selected), on the current File Server. The standard printer server banner and end-text characters (as set up by the system manager) will be printed.

*PRINTOUT therefore allows a program that generates printed output, to run even if the printer is currently busy. Instead of sending output directly to the printer (by *FX5,4 or *PS followed by <ctrl-B>), the user can spool the screen output to a file by typing *SPOOL <file specifier>, and terminating the file by typing *SPOOL on its own. When the printer becomes free, the user should then type *PRINTOUT <file specifier>, which will send it the contents of the file. It is recommended that users run *PUTGET before running *SPOOL -- see notes under *SPOOL in this Section.

Another advantage of *PRINTOUT over standard printing is that the user is informed if there is any problem. Sending <ctrl-B> (to start printing through the network) while the printer is busy, will cause the user's BBC Micro to hang for about 30 seconds, before coming back with **Not listening**. The error messages generated by *PRINTOUT are more informative as well as appearing immediately.

There is a possible hazard with *PRINTOUT, if your printer does an automatic line-feed whenever it receives a carriage return character (ASCII value &0D). When doing normal printing from a BBC Microcomputer, line-feed characters (ASCII &0A) will be filtered out by the BBC Micro (unless this is changed with the *FX6 command, as described in the BBC User Guide, page 408). *PRINTOUT does not do this filtering, which means that text containing line feed characters will be double spaced on a printer which does auto-line-feed.

The solution is to turn off the auto-line-feed option on the printer, and type *FX6,0 at each BBC Micro (this can be done in a boot file whenever a user logs on). The behaviour of *PRINTOUT will then be consistent with normal printing.

Note also that the text files produced by many word processors do not contain line-feed characters. If *PRINTOUT is to be used with such text files, the user should start a spool file (as above), then print the text to the computer screen, then use *PRINTOUT with the spool file.

Examples:

***PRINTOUT TEXTFILE**

will send the file TEXTFILE to the printer.

***SPOOL LISTING
LIST.**

.

(BASIC listing on screen)

.

.

.

***SPOOL
*PRINTOUT LISTING**

will send a listing of the BASIC program in memory, to the printer.

Likely Errors :

Error 09 (9) Printer busy with station nnn (xxxx) -- if the printer is not currently available.

Error 0A (10) Not authorised to use printer -- The system manager can set up a printer, so that only users with access to a certain account number can use it.

Error BD (189) Insufficient access -- The user must have read access to the specified file.

Compatibility Notes :

Not supported by Acorn systems.

***PS**

Command Type : Transient program.

Syntax :

***PS <station number> | *PS | *PS <printer name>**

Action with Wild Cards : Wild cards prohibited.

Description :

The Econet software in the BBC Micro assumes that the network printer server has number 235 unless specified otherwise by this command. The SJ Research File Server contains a printer server, so if the File Server is station 254 and the built-in printer server is used, the *PS command will be necessary to get any output. This program will also perform the call ***FX5,4** to select the network printer. The program can also have the effect of a ***FX6,0** call if the printer in your network is not set to do automatic line feeds. (The system manager will configure the program to suit the printer.)

If there are several stations with printer servers on the network, the ***PS** program can be used to search for an unoccupied printer server. Typing *PS on its own will select the first free printer available, or typing ***PS <printer name>** will select a printer of a certain name.

The system manager will allocate specific names to particular types of printer, and the use of *PS <printer name> will search for a printer with that particular name. There can be several printers with the same name, but the system manager would only give the same name to identical printers. With current versions of software you must be logged on to a File Server to be able to use ***PS <printer name>**

After *PS (either with a name or with no argument), the system will display the station numbers of all the printer servers that responded, followed by **Station nnn selected**, or **No station responding** if a corresponding printer cannot be found.

Note that *PS <printer name> or *PS (without an argument) will work only if the printer servers (in BBC Microcomputers) are equipped with NFS 3.6 or later software. With NFS 3.34 in the printer server, the search will not work, but the use of *PS <station number> will function normally.

Examples :

There may be a daisy-wheel printer with name **DW**, and some Epson dot-matrix printers with name **EPSON**. To find a free dot-matrix printer, the user could type

***PS EPSON**

The system would then reply either (for example)

Station 235 busy
Station 234 ready
Station 234 selected

or

No station responding

Likely Errors :

There are no errors specific to this program. If the printer server station is specified explicitly, no check is made on its existence until printing is attempted, when **Error A2 (162) Not listening** will occur if the printer server is either busy or non-existent.

It is possible that any program calling *FX5 may not work if the local printer buffer is not empty -- if, for example, a local printer has been used but had not completed the job. If *PS appears to hang up, press <escape> <Ctrl-B> <escape>, then try *P5

again.

Compatibility Notes :

Supported by Acorn systems, but only the *PS <station number> or *PS (no argument) versions.
Note also that Acorn versions of the *PS command may work differently.

***PTIME *PDATE *PDATE2**

Command Type : Transient programs

Syntax :

***PTIME | *PDATE | *PDATE2**

Action with Wild Cards in File Name : Not applicable.

Description :

These programs print the time of day or the date without preceding or following new line characters. They are intended for use from a BASIC or other program, and allow the user to enclose the output in text if required, as shown in the example below.

The printed format from PTIME is

hh:mm

The printed format from PDATE is

dd/mm/yy

The printed format from PDATE2 is the date in text form, for example **16th February, 1984**

Examples :

```
340 PRINT "The time is now";
```

```
350 *PTIME
```

```
360 PRINT "hours past midnight"
```

Note that, whilst lines 340 and 350 may be combined with a colon, the following text must be on its own line if PTIME is called in this way. This is because BASIC (or any other language system) passes the entire remainder of the line to the operating system when it meets a "*" character. (In BASIC 2, you could use OSCLI("PTIME") in place of *PTIME, to avoid this problem).

Likely Errors :

There are no errors specific to this program. A nonsensical time output implies that the real-time clock has not been set, or that the File Server has been switched off for several months, so that the clock battery has run down. Ask your system manager for help.

Compatibility Notes :

Supported on Acorn systems. Level 2 File Servers have only the date; level 3 File Servers have both date and time, and it uses the interval timer function in the BBC Microcomputer thereafter -- there exists the possibility that this timer could lose time under certain circumstances.

***PUTGET and *PUTGET2**

Command Type : Utility program -- must be run before a loading user program.

Syntax :

***PUTGET**
***PUTGET2**

Action with Wild Cards in File Name : Not applicable.

Description :

PUTGET is a program which resides in memory (it reserves some private workspace for itself rather like a sideways ROM) and implements local buffering of random access operations to files. That is it 'packages up' **BGET** and **BPUT** calls into blocks of 64 bytes, then uses the OSGBPFB filing system call. This results in a speed increase of roughly 64 as it takes almost as much time to read or write a single byte as to read a block of 64 bytes, due to network and File Server overheads.

To activate simply type ***PUTGET**, and a banner will be printed. PUTGET will remain active (even over BREAK) until you specifically de-activate it. The most noticeable effect of PUTGET is to move the BASIC value PAGE up by three pages, and because of this it will corrupt any BASIC program that is in RAM. Users should therefore be sure to run PUTGET before loading any BASIC programs.

The use of PUTGET will significantly speed up programs that use the BASIC commands **BPUT# BGET# INPUT# PRINT#**, and the OS commands ***EXEC *SPOOL** and the DFS Utilities (if fitted) ***TYPE *DUMP *LIST *BUILD**.

Running PUTGET twice will produce the message **PUTGET already active** -- this message is just printed and does not cause an error.

A special version **PUTGET2** is provided for use with word processors such as WORDWISE. This version loads into the RS423/cassette tape filing system buffer, and provides more space for text. Note however that this is illegal workspace, and PUTGET2 corrupts these buffers. PUTGET2 should therefore not be used except with Wordwise etc., and especially not if it is intended to use the serial input or output, or the cassette tape.

PUTGET can buffer up to two files: one of which must have been opened for input or update, the other for output. Any further files are passed straight through to the NFS and will not be buffered.

PUTGET only allows access to files on a sequential basis, that is there is no facility for using PTR# and EXT# although EOF# of course still works. If PTR# and EXT# need to be used, or if more memory is required (PUTGET takes up 3 pages of RAM, i.e. it raises the BASIC variable PAGE by &300) then PUTGET must be de-activated. To do this type either <Ctrl-Break> or *FX247<ret> then BREAK.

In detail, what happens is that PUTGET is a short loader program that executes at &A00. This loads the file PRL.PUTGET, which is the actual PUTGET code in relocatable format (this must be in the library in the sub-directory PRL) at the operating system 'high water mark' (OSHWM), and then relocates and executes it. This code intercepts various vectors in page 2 (OSARGS to OSFSC and OSBYTE) and the BREAK key intercept, implodes the font (using *FX20,0) and raises OSHWM by 3 pages. It will then re-explode the font and finally re-start the current language (which in the case of BASIC is needed to re-initialise PAGE to the new value of OSHWM).

Examples :

***PUTGET**

Ideally this should be run at the beginning of every session. perhaps by placing it in

a !BOOT file, either for the user, or in the library. See under *OPT4 for details.

Likely Errors :

There are no error messages specific to PUTGET, except for the non-fatal **Putget already active.** PUTGET will produce this message and cause no further action.

PUTGET loads a file called PRL.PUTGET from the library, and that the error **PUTGET not found** or **PRL not found** will occur if this file or directory PRL are absent.

Note that running PUTGET after LOADING a BASIC program will give nothing if the user attempts to list or run the program.

Compatibility Notes :

This utility will run identically with Acorn File Servers, and its use is recommended with them as well as with SJ Research File Servers.

***RENAME**

Command Type : File Server command

Syntax :

*RENAME <old general specifier> <new general specifier>

Action with Wild Cards in File Name : Occurs on every match. See below for full details.

Description :

This command changes the name of an existing file or directory. The user must own the item (i.e. he must have access to the main or auxiliary account of the item). An attempt to rename an item so that it would have the same name as an existing file (or directory) will give the error message **Already exists**

A file (or directory) may be moved into another directory using this command, but the user must be an owner of the new directory as well as of the item being renamed (even if he is renaming the item solely within one directory). Note that he does not have to be an owner of the directory from which he is moving a file.

An attempt to rename a directory as a sub-directory of itself will give rise to the error **FS unusual error 07**.

In a system with more than one disc, it is not permitted to use this command to move files from one disc to another. (The utility programs COPIER or MULTICOPY should be used for this).

Wild card characters are permitted in the file specifiers, and will cause the renaming to occur on all items that match the wild card specifier. The same wild card character(s) must appear in the old and new specifiers.

Examples :

***RENAME MYPROG \$.PROJECT.MYPROG**

will move MYPROG into directory \$.PROJECT without changing its name. The user must be an owner of \$.PROJECT for this to be allowed.

RENAME PROG* DATA

will rename PROG1 as DATA1, PROGDEMO as DATADEMO and so on. ***RENAME A#PP* DATA#PP*** will rename AZPP1 to DATAZPP1, AMPP to DATAMPP and so on, but not A1PPINFO to DATA1PPINFO, because the new name has more than 10 characters and so would cause an error -- under the wild card rules, this item would be passed over (see Section 6.3). Note also that the wild cards must correspond in the old and new specifiers.

Likely Errors :

Error 07 (7) Circular RENAME -- caused by an attempt to rename a directory as a sub-directory of itself.

Error B0 (176) Renaming across discs -- It is not possible to transfer an item from one disc to another using *RENAME.

Error BD (189) Insufficient access -- The user does not own either the file being renamed, or the directory in which he is trying to rename it.

Error CC (204) Bad wildcard -- The number and types of wild card characters must match in the *RENAME command.

Error C4 (196) Already exists -- from an attempt to rename an item to have the same name as an existing item.

Compatibility Notes :

Supported by Acorn systems. Note that you may not rename directories on an Acorn

File Server, and that wild cards are not allowed.

***RUN**

Command Type : BBC Microcomputer operating system command

Syntax :

***RUN <file specifier> | */<file specifier> | *<file specifier>**

Action with Wild Cards in File Name : Occurs on first match (alphabetically).

Description :

This commands will load and run a machine code program at a location specified when the file was saved (see under *SAVE).

The action taken by the file server is to search through the currently selected directory for the file specified. If it is not found, the file server will search through the currently selected library (see under *LIB command), and will load the file from there if found.

The command */<file specifier> is exactly equivalent to *RUN <file specifier>. If there is no sideways ROM or filing system command with the same name as <file specifier> then the version *<file specifier> may be used. If there is such a command, then the full version *RUN (or */) must be used.

Examples :

***PUTGET**

runs the program PUTGET from the currently selected directory (CSD) or, if it is not found in the CSD, from the library

***RUN DELETE**

will run a program called DELETE from the CSD or library. This form of command avoids the File Server command *DELETE

***/DELETE**

is identical to the example directly above.

Likely Errors :

Error B5 (181) xxxx is a directory -- An attempt has been made to *RUN a directory.

Error BD (189) Insufficient access -- The user must have access R to the file.

Error D6 (214) Not found -- The file does not exist.

Compatibility Notes :

Supported on Acorn systems.

SAVE

Command Type : BASIC keyword

Syntax :

SAVE "<file specifier>" | SAVE <string variable>

Action with Wild Cards in File Name : Wild cards prohibited.

Description :

SAVE is a BASIC command, that causes the BASIC program currently in memory to be saved, with a file-name equal to the string immediately following the SAVE command.

The user should take care to type OLD after pressing <Break> on the BBC Microcomputer. If the program is saved after <Break> without doing this, then a trivial file of two bytes length will be created, over-writing any existing file of that name.

The system manager has an option available to prevent users SAVEing files of less than 16 bytes in length, to prevent the above problem. An attempt to save a short file will give the error **Too short**.

Examples :

SAVE "Prog27"

will save the current program as Prog27

A\$=" My Prog"

followed by

SAVE A\$

will save the current program as **MyProg**.

Likely Errors :

Error 06 (6) Too short -- caused usually by doing SAVE after pressing <Break>. The system manager has to set a special option before this error will be caused, and its occurrence can be changed using *ENABLE SAVES or *DISABLE SAVES.

Error B5 (181) xxxx is a directory -- caused by attempting to create a file with the same name as that of a directory.

Error BD (189) Insufficient access -- caused if the user is not an owner (i.e. does not have access to the main or auxiliary account) of the directory in which the new file is to be created.

Error C2 (194) Already opened by xxxx If this or another user has this file open for reading or writing, then it cannot be deleted by creating a file with the same name, until it has been closed.

Error CC (204) Bad name -- if the file name contains illegal characters (e.g. \$ % . ^ : except in contexts where they are permitted)

Error CC (204) Bad wildcard -- use of * or # in the file name.

Error C3 (195) Entry locked -- It will be necessary to use the *ACCESS command to unlock the file before it can be deleted by creating a file with the same name.

Compatibility Notes :

Supported on Acorn systems.

***SAVE**

Command Type : BBC Microcomputer operating system command.

Syntax :

```
*SAVE <file specifier> <base addr.> <top addr.> |  
*SAVE <file specifier> <base addr.> <top addr.> <execute addr.> [<load addr.>] |  
*SAVE <file specifier> <base addr.> + <length> |  
*SAVE <file specifier> <base addr.> + <length> <execute addr.> [<load addr.>]
```

Action with Wild Cards in File Name : Wild cards prohibited.

Description :

*SAVE creates a new file, containing a copy of the specified memory area.

The base address is the address in memory from which the copying will start, and the top address the address next above the last one that will be copied. If the length is specified after a + sign, the top address will be found by adding the length to the base address.

When a file is saved, the values of the load address and the execute address are stored in the appropriate directory. The execute address is the address in memory that the computer will jump to after loading the file in a *RUN command, and the load address is the base address for the file when loading with a *RUN or *LOAD command.

If the load address is not specified, then it will be taken as equal to the base address. If the execute address is not specified, it will also be taken as equal to the base address. If you wish to specify the load address you must also specify the execute address.

Examples :

```
*SAVE IMAGE 2000 280E
```

will save the area of memory between 2000 and 280D (inclusive) into a file called IMAGE.

```
*SAVE COMP 2000 295E 3020 3000
```

will save between 2000 and 295D into a file called COMP. When *COMP is typed, the file will be loaded at address 3000, and the system will then jump to address 3020.

Likely Errors :

Error 06 (6) Too short -- caused by an attempt to save a file of less than 16 bytes length. This error is caused only if the system manager has set a special option for a user, and its occurrence can be changed using *ENABLE SAVES or *DISABLE SAVES.

Error B5 (181) xxxx is a directory -- caused by attempting to create a file with the same name as that of a directory.

Error BD (189) Insufficient access -- caused if the user is not an owner (i.e. does not have access to the main or auxiliary account) of the directory in which the new file is to be created.

Error C2 (194) Already opened by xxxx If this or another user has this file open for reading or writing, then it cannot be deleted *by* creating a file with the same name, until it has been closed.

Error CC (204) Bad name -- if the file name contains illegal characters (e.g. \$ % . ^ : except in contexts where they are permitted)

Error CC (204) Bad wildcard -- use of * or # in the file name.

Error C3 (195) Entry locked -- It will be necessary to use the *ACCESS command to unlock the file before it can be deleted by creating a file with the same name.

Compatibility Notes :

Supported on Acorn systems.

***SDISC**

Command Type : File Server command

Syntax :

***SDISC <disc name>**

Action with Wild Cards in Disc Name : Occurs on first match in order of disc drive numbers.

Description :

This command is equivalent to ***DIR :<disc name>** Note that the subsequent use of *DIR (without an argument) will return to the initially selected disc.

Examples :

***SDISC MAIN2**

will select as CSD the root on disc MAIN2.

Likely Errors :

Error CC (204) Bad name -- if the disc name contains illegal characters.

Error D6 (214) xxxx not found -- if the disc of that name is not present in the system.

Compatibility Notes :

Supported in Acorn systems, but note that *SDISC is equivalent to *DIR :<disc name>.<user identifier>, and that a subsequent *DIR will remain on the new disc on Acorn File Servers.

***SPOOL**

Command Type : BBC Microcomputer operating system command.

Syntax :

***SPOOL <file specifier> | *SPOOL**

Action with Wild Cards in File Name : Wild cards prohibited.

Description :

This command causes the specified file to be created (deleting any existing file of the same name), and for all subsequent output to the screen of the BBC Microcomputer to be written to it. A common application is to make text files out of BASIC programs, or as a simple way of making program generated text. Some examples are given below.

The file is closed by running ***SPOOL** without a file specifier. Note that the file handle in the BBC Microcomputer will be lost if <Break> is pressed, and so the spooling will stop. The File Server will still have the file open however, so under these circumstances the file will have to be closed either with **CLOSE#0** or by logging off with ***BYE**.

In detail, ***SPOOL** performs an OPENOUT operation, and then uses the BPUT call to send all text from the BBC Microcomputer screen, to the specified file. This operation is fairly slow over the Econet network, so it is recommended that the utility program ***PUTGET** is run before using ***SPOOL** (see description under ***PUTGET**)

Examples :

This type of program allows the user to create a file !BOOT :

```
10 *SPOOL !BOOT
20 PRINT "*TV0,1"
30 PRINT "*" | THIS IS THE MESSAGE OF THE DAY"
40 PRINT "*PUTGET"
50 *SPOOL
```

The sequence below will produce text from a BASIC program, which can then be edited by a word processor. The ***EXEC** command (q.v.) can be used to turn the edited text back into a BASIC program .

```
*SPOOL TEXTFILE
LIST
```

```
*SPOOL
```

Likely Errors :

Error B5 (181) xxxx is a directory -- caused by attempting to create a file with the same name as that of a directory.

Error BD (189) Insufficient access -- caused if the user is not an owner (i.e. does not have access to the main or auxiliary account) of the directory in which the new file is to be created.

Error C0 (192) Too many files open -- There is a limit to the number of channels available, normally 5 or 6.

Error C2 (194) Already opened by xxxx -- If this or another user has this file open for reading or writing, then it cannot be deleted by creating a file with the same name, until it has been closed.

Error CC (204) Bad name -- if the file name contains illegal characters (e.g. \$ % . ^ : except in contexts where they are permitted)

Error C3 (195) Entry locked -- It will be necessary to use the ***ACCESS** command to unlock the file before it can be deleted by creating a file with the same name.

Compatibility Notes :
Supported on Acorn systems

***STATEMENT**

Command Type : Transient program

Syntax :

***STATEMENT**

Action with Wild Cards in File Name : Not applicable.

Description :

This program gives a list of all the accounts that the user has access, with the associated credit balances.

The credit balance of an account is in units of 1 kilobyte (K), and represents the space available for files (or directories) with the corresponding account number. If a user has access to more than one account, he may move files from one account to another by use of the *ACCOUNT command (q.v.). The account of a file or directory also determines the ownership of that item -- if a user has access to the account of an item, then he owns that item. A full explanation is given under *ACCOUNT.

If there is more than one disc in the system, then *STATEMENT will give a separate list of accounts for each disc. The user will however only be able to create files on discs on which there exists a directory with account to which he has access.

Transient programs run in a reserved area of workspace in the computer, and will not corrupt programs in the main memory.

Examples :

***STATEMENT**

The system will reply (for example)

Disc 0	
Account	Balance
88	252k
89	66k
90	bankrupt

Disc 1	
Account	Balance
88	45k
89	512k
90	511k

Likely Errors :

There are no errors specific to this program.

Compatibility Notes :

Not supported on Acorn systems, since these do not run an accounting system.

***STATIONS**

Command Type : Transient utility program

Syntax :

***STATIONS [<network number>]**

Action with Wild Cards in File Name : Not applicable.

Description :

This utility displays a list of all machines connected to the Econet network, that are currently switched on. Each line is of the form

<station number> <machine type>

The machine types include

- SJ Research File Server
- BBC micro
- System 5
- System 3/4
- Z80 CP/M

*STATIONS will not show the number of the computer running this program (the program *CV will do this if required). It will also not show the numbers of any computers in which the network interface has been disabled -- the program NETMON and some computer games do this.

If the installation has multiple networks connected by bridges, then *STATIONS with no argument will display only active stations on the same network. If *STATIONS is followed by a number between 1 and 255, then the program will look for a bridge joining to the network of this number, and will list the stations active on that network.

Examples :

***STATIONS**

to which the system will reply (for example)

Station	Type
254	SJ Research File Server
87	BBC micro
2	BBC micro

***STATIONS 3**

will search for network number 3, and will produce a list similar to above, of stations on network 3.

Likely Errors :

There are no errors specific to this program

Compatibility Notes :

Supported by Acorn systems.

***TIME**

Command Type : Transient utility program.

Syntax :

***TIME**

Action with Wild Cards in File Name : Not applicable.

Description :

This utility prints out the time and date, from the a 'real time clock' contained to the file server. The form of the output is given in the example below.

There are alternative versions of this program suitable for incorporating into user programs. These are ***PTIME *PDATE *PDATE2** -- all three produce output without any <return> characters around the text. See under PTIME. for details. There is also a program ***GTIME** which inserts the time in a machine readable form into the BBC Microcomputer.

Examples :

***TIME**

The system will reply (for example)

The time is 11:10:46 on 08/03/84

Likely Errors :

There are no errors specific to this program. A nonsense time output implies that the real-time clock has not been set, or that the File Server has been switched off for several months, so that the clock battery has run down. Ask your system manager for help.

Compatibility Notes :

Supported on Acorn systems. Level 2 File Servers have only the date; level 3 File Servers have both date and time, and it uses the interval timer function in the BBC Microcomputer thereafter -- there exists the possibility that this timer could lose time under certain circumstances.

***TYPE**

Command Type : Transient utility program

Syntax :

***TYPE <file specifier>**

Action with Wild Cards in File Name : Occurs on first match (alphabetically).

Description :

This program opens the specified file, and prints it as a text file on the screen of the computer. Tab characters (ASCII value 9) will be replaced by sufficient spaces to bring the next character to a multiple of eight character spaces from the left margin.

Note that if the file contains 'control' characters (ASCII characters 0-31), then these not be printed to the screen. (This is because they would affect the screen directly, and probably make a nonsense of the output -- for example ASCII 21 <Ctrl-U> will turn the screen off altogether). The only exceptions to this are the characters &09 (tab), &0C (page feed) and &0D (new line), which will be printed. Characters of ASCII value greater than &80 ('top bit set' characters) will also not be printed.

This program uses the multiple byte transfer OSGBP call, and so will run considerably faster than for example the version of *TYPE that is contained in the DFS ROM. If DFS is fitted to the computer, use ***/TYPE** to be sure of running the network version. (See under ***RUN** for full details of */).

Examples :

***TYPE FILE1**

will print FILE1 to the BBC Microcomputer screen. A printed copy could be made at the same time by typing <Ctrl-B> before the command, to turn on the printer, and <Ctrl-C> at the end to turn it off. See Section 6.5 about printing through the network.

Likely Errors :

There are no errors specific to this program. However, it performs an OPENIN call, and so can cause all the same errors that the OPENIN would.

Compatibility Notes :

Supported by Acorn systems.

***USERS**

Command Type : Transient utility program

Syntax :

***USERS**

Action with Wild Cards in File Name : Not applicable.

Description :

This utility prints a list of all the users currently logged-on to the file server, the station numbers at which they are logged-on, and any system privilege that they may have.

If a user is logged on at several stations, then he will appear several times in the list. The list is re-ordered every time a filing system operation occurs, so that the user who most recently performed a filing operation will be at the top of the list -- this will, of course, be the user who ran *USERS from the network.

It is worth noting that a command like ***NOTIFY <user identifier>** will only notify one occurrence of the user. The one which will be notified is the same as the one that appears first in the list produced by this program, which should be the last computer at which the specified user performed a filing operation.

Note that if a user was logged-on, and switches off his machine, he will still appear in *USERS until someone else turns that machine back on and logs-on from it.

One of the users listed may be *SYSTEM* at station 0. This occurs if a user has run *PRINTOUT, and the system is printing this job. This user does not have system privilege.

Examples :

***USERS**

to which the system will reply (for example)

Station	User Id.	Privilege
026	SYST	System
043	FRED	
000	*SYSTEM*	
218	JOHN	

Likely Errors :

There are no errors specific to this program.

Compatibility Notes :

Supported on Acorn systems, but the user list is not sorted in the same way, and is essentially random in order.

***VERS**

Command Type : Transient utility program

Syntax :

***VERS**

Action with Wild Cards : Not applicable.

Description :

This program displays the current version number of the File Server. This will be of the form

S3 Research File Server ver n.nn / <file server type>

Examples :

***VERS**

to which a typical reply will be

SJ Research File Server ver 0.67/HDFS

for a hard disc File Server.

Likely Errors :

There are no errors specific to this program.

Compatibility Notes :

Supported by Acorn systems, but the displayed number will be different.

7. An Application Note for Econet Programming on the BBC Microcomputer

Topics covered:

- The NFS ROM
- OSARGS
- Printing
- File handling
- Econet Workspace
- Use of OS workspace (or, How to make games work)
- The program NETMON
- Basic Guide to Econet Line Protocols

Glossary of terms and abbreviations used in this section:

- AUG Advanced User Guide for the BBC Microcomputer - Bray, Dickens & Holmes.
- ESUG The Econet System User Guide (The greyish book).
- EAUG Econet Advanced User Guide (The black book with sky and clouds on it).
- UG BBC Microcomputer User Guide - beware, details of OSFIND etc are WRONG.
- OSHWM Operating System High Water Mark -- A system variable, to which BASIC sets the value of PAGE.

The value of OSHWM depends on how many filing system ROMs are present in a BBC machine. (Languages do not take up any room, except when they are active). Here is a list of some common configurations:

ROMs fitted	value of OSHWM
No ROMs	&0E00
Disc	&1900
Econet	&1200
Disc & Net	&1B00
Teletext	&2200
Teletext & Disc	&2400
Teletext & Econet	&2400
Teletext, Disc & Net	&2600

7.1 The NFS ROM

Two common versions of the NFS ROM exist. The one you will probably have is NFS 3.34 in EPROM: if you are lucky you may have a DNFS (in ROM), and if you know the right person, an NFS 3.6 EPROM which is a DNFS ROM 'cut in half'. Acorn is also developing an 'Advanced NFS', (ANFS) but it is not known when this is to be available. The main feature of ANFS is that it does its own 'local buffering' - see *PUTGET.

To find out which version you have, type ***HELP**. It is possible to read the NFS type using OSARGS (see below), which is useful for version-independent programming.

Whenever a reference is made to NFS 3.6, this also includes the DNFS ROM, as the Econet code is identical in both.

7.2 OSARGS

See EAUG p. 40 for preference, also ESUG p.36 and AUG p.337.

Documentation relating to 'Return command line address' call, which is used to access parameters for * commands, eg

***PS DAISY**

where PS is a loaded program being run, requiring access to the string **DAISY**.

OSARGS call address &FFDA with A=0, Y=0, X pointing to 4 locations in zero-page.

NFS 3.6

The address returned is the address of the first non-space character after the filename that is being RUN. This is the 'correct' value and the one returned when using a DFS.

NFS 3.34

The address returned is incorrect but reasonably well defined: it points to the name of the file being run, except in the case of */ <filename> where the address points to the first character after the slash: leading spaces are not stripped.

Do not assume that registers are preserved on exit from this call, even though the DFS manual says X and Y are preserved. In fact X is INC'd on exit from this particular brand of OSARGS call on the NFS.

Handling of this call should proceed as follows:

- a) Find out NFS version number by using OSARGS with Y=0, A=2.
NFS 3.34 returns A=2, NFS 3.6 returns A=1, ANFS also returns A=1.
If NFS 3.34 then Continue....
- b) Skip leading spaces.(We are now pointing at the filename being RUN)
- c) Skip non-space characters
- d) Skip spaces (now pointing at right item)

7.3 Printing

With NFS 3.34 printing anything other than straight text is not guaranteed. This is because it is not possible to send the characters 2 and 3 to the printer without them being interpreted specially, even by using VDU 1. There is another problem with the NFS inserting extra 'null' characters into the print stream, this also being dependant on whether printing was started with a CTRL-B typed in at the keyboard or using a VDU 2. (Don't ask me why but it is believed that VDU 2 is safer).

With NFS 3.6 a new print protocol has been devised which allows any character to be sent to the Print Server, thus allowing graphics dumps to be printed.

When using the Econet Print Server it is necessary to understand the concept of logging-on and -off (The same as with an FS). Basically VDU 2 logs on, and VDU 3 logs off: on non-spooling Print Servers (eg the Acorn PS ROM, some SJ File Servers) logging-on has the effect of 'locking-out' other users so that they get a 'Not listening' error. Logging-on also has the effect of printing a banner, including possibly time of printing and the user's name, whilst log on & off flushes the print buffer and sends a form-feed. Therefore these two commands should **not** be used to enable/disable output to a printer (when some characters must go to screen but not the printer), even though this works OK with a local printer.

To enable/disable printer output after a VDU 2, use *FX3 4 to disable the printer and *FX 3 0 to enable it again. See page 119 of the AUG or page 422 of the UG.

7.3.1 PS Status Enquiry call

It is possible to find out the state of a print server on the network, and also to find out what printers are available in general. The utilities ***PS** and ***POLLPS** (of which only the former is available currently) can be used, and I have included the low-level interface for completeness:

Client to PS

Port: &9F

6 bytes of data - ASCII string as described below

2-byte Function code (=1)

PS to Client (Note that the RxCB should be set up to receive 3 bytes of data)

Port: &9E

1 byte status:

0 - Ready

1 - Busy with station mm.nn

2 - Offline

2-bytes: station number

The data sent to the PS is in the form of an ASCII string padded with spaces describing the name of the printer. 'PRINT ' and 'AUTO ' are two defined names: depending on how the PS is set up others may be used.

As an important corollary, sending the Status Enquiry call as a 'Broadcast' message will cause all PS's that recognise the string to reply. This is what the utility *PS uses.

7.4 File handling

As has been strongly emphasised to you by now, file access using BGET#, BPUT# (and therefore also INPUT#, PRINT#) is S L O W. The remedy is to use OSGBPb (often pronounced 'Ozheebeegeebee'). One of the problems about using this call is the lack of documentation, and also the fact that the cassette filing system (CFS) does not support OSGBPb, but only OSBGET and OSBPUT: this means that a lot of commercial software packages, wishing to be compatible with Tape FS, use BGET and BPUT. The only solution here is to use PUTGET (q.v. §6.6).

For those interested in writing their own file handlers, I have prepared a useful list of references and routines.

OSGBPb documentation can be found in:

Acorn DFS manual (The yellow and green book) p. 73
ESUG p. 37
EAUG p. 43
AUG p. 339

The most important points to remember are these:

OSGBPb call address is &FFDI, and on entry:

YX points to a 13-byte control block (even if you are not using the given pointer) A is a reason code:

1,2 Put bytes (1 uses the given pointer)
3,4 Get bytes (3 uses the given pointer)

On exit:

All except the first byte of the control block are updated.

DO NOT PUT THE CONTROL BLOCK IN PAGE ZERO. There is a bug in NFS 3.34 such that OSGBPb does not work (at all) if you do this.

From BASIC it is convenient to have a procedure such as:

```
DEFPROCgbpb(A%,handle%,buffer%,length%,offset%)
  LOCAL X%,Y%
  X%=gb%
  Y%=gb% DIV 256
  ?X%=handle%
  X%!1=buffer%
  X%!5=length%
  X%!9=offset%
  CALL &FFD1
ENDPROC
```

with a statement of the form:

```
DIM gb% 12      (note gb%?0 and gb%?12 are both available so this has allowed for 13 bytes of memory)
```

somewhere near the beginning of the program.

As an example, the call PROCgbpb(4,F%,BUFF%,100,2000) would load 100 bytes from offset 2000 in the file whose handle is F% to location given by the value of BUFF%. Normally you would have a DIM BUFF nnnn statement in your program to allocate sufficient space for use by this procedure.

7.4.1 OPENIN

The other most important aspect of file handling on the network is the difference between OPENIN and OPENUP, and the fact that BASIC-I's OPENIN is in fact interpreted as an OPENUP. If you have BASIC-I you will have to write a special function for OPENIN, viz.

```
DEFFNopenin($ob%)  
  LOCAL A%,X%,Y%  
  A%=&40 (OPENIN type)  
  X%=ob%  
  Y%=ob% DIV 256  
=(USR&FFCE) AND 255
```

where, like gb%, ob% has been DIMENSIONED and is an 'OPENIN buffer', and should be of the order of 80 bytes to enable it to accomodate a reasonably long filename (see below).

(See AUG p.342, ESUG p. 34, EAUG p. 40. IGNORE the UG.)

Thus instead of F%=OPENIN("file") do F%=FNopenin("file"). On BASIC-II this is not necessary.

7.4.2 File names

Although individual components of an Econet file name may not be longer than 10 characters, it is quite possible to have a compound file name (eg \$.JOHN.BBCprogs.....2ndrev.OLD.developmnt. Addition) of almost infinite length. A compromise length of 80 characters should be used as a minimum. Even a DFS file name can be up to 12 characters long (:2.R.LONGEST), requiring 13 bytes of storage including the CHR\$(13) on the end. Programs which disallow filenames over 7 characters long should not be considered to be programs at all.

7.5 Econet workspace

When writing * commands it is desirable to allow them to run in areas of memory which do not corrupt BASIC programs and other valuable user- or system data. The most obvious way of doing this with Econet-specific commands is to use the Econet workspace (Public area) which comprises pages &E and &F. When writing Filing-system independant commands use the area mentioned in note b) below.

As a general guide the following areas are safe:

E10..E1D	OK
E1E..E22	Corrupted during loading
E23..E2F	OK
E30	? DNFS stores file name
E31..EFF	OK
F00-F03	used by * commands, OSFILE etc.
F03..F04	must be zero
F05..F08	Corrupted during loading
F09..FOC	must contain the 32-bit execute address
F0D..FDC	OK

Notes:

- F00 is used as a buffer for all FS-related commands eg *I AM , *<filename>, OSFIND etc.
- It seems now to be an accepted convention that if you can't get * commands to work in the Econet workspace, then pages 9 and 10 (ie &900..&AFF) should be used.

7.5.1 Zero-page workspace

If you are writing your own * commands, and need some zero-page workspace, it is important to use the correct area. There is an area specifically reserved for such 'Operating System' commands, and this is at locations &A8..&AF. See AUG p. 268.

NB it is not good enough to use the 'spare' BASIC workspace &70..&8F, as there is no reason why a user may not run any * command from within any language, not necessarily BASIC.

7.6 Use of OS workspace

The area of memory below PAGE (or to be more precise OSHWM) is reserved for use by the OS and any filing systems that may be in your machine. Interfering with this memory can have unexpected and disastrous effects. However, many games and other (illegal, in the bad programming sense of the word) programs use RAM below PAGE, sometimes as low as &400 (an admittedly exceptional case).

On the Econet this is particularly nasty as the NFS uses NMI which will interrupt ALL machines whenever any net traffic occurs. This results in machines crashing when plugged into the network. In fact it is not only the NMI workspace which is corrupted during an interrupt but also the NFS Public workspace (pages &E and &F) and the NFS Private workspace, which effectively means any part of RAM below OSHWM.

The solution is for the user himself to 'claim' the NMI workspace, which has the effect of disabling the Econet altogether. It is best documented in the AUG pp. 320.

To claim NMI workspace use OSBYTE &8F, viz.

```
A%=&8F (sideways ROM call)
X%=&C (the 'claim' call)
Y%=&FF (as it instructs you to do)
CALL &FFF4
```

Of course the main disadvantage of disabling the Econet is that you cannot use any of its facilities while the program in question is running. If you are writing your own software, disabling the Econet should only be considered as a last-ditch option. Educational users will be particularly annoyed as the majority have networks and may wish to use network related utilities with such software - *SHOW would be an example: it is a program which sends all screen output down the network, to anyone who is *WATCH-ing.

In addition, overwriting filing-system workspace can often cause the micro to crash on pressing BREAK. The NFS has been known not to recover even after a Ctrl-BREAK - a power-off was necessary!

7.7 Description of the Program NETMON

NETMON provides a means of continuously monitoring the network at a very low level. It displays the data bytes as they are sent down the network and also some status information, particularly useful in the debugging of network software, and networks in general.

In order for NETMON to be of much use one has to be familiar with the Econet line protocols: a introduction is given below, and a comprehensive and detailed description is given in section 7.8. There is also some NETMON-specific debugging output.

*NETMON loads some very special code which runs the network hardware directly. This effectively removes it from the network, as far as other machines and the program ***STATIONS** is concerned. It is wise after running NETMON to power-off before attempting to use it for normal programming purposes.

The program prints:

```
ECONET MONITOR xxx  
1E
```

where xxx is the station number of the monitor machine. Ignore the '1E' - even I don't know why it is printed. Monitor output can at any time be stopped by pressing the space bar (ctrl-shift functions as normal).

Data bytes as they are sent on the network are printed in hex. There are various statuses that are printed and these are:

<space>

Address present.

Indicates that the next byte is an address byte, which is always the first of a packet. Packets are therefore always separated by spaces.

i <newline>

Idle detected.

Occurs between any two Econet messages. Stations wishing to transmit must always wait for an idle condition on the line (a sequence of 15 one's) before enabling their line drivers. A long string of i's is caused by network hardware problems, usually either poor wiring, or one or more blown SN75159 line driver ICs.

v Frame valid.

Occurs just before the last byte of a packet, indicating that the CRC was valid.

e CRC error.

This can be caused by two stations transmitting simultaneously, or by noise getting into the network, or by an intermittent network connection, or a bad econet lead anywhere one the network, or....

o Data overrun error.

At clock speeds above 160KHz or thereabouts the monitor cannot keep up with the data rate. However, BBC machines are quite capable of running up to about 235Kbaud (2nd processors only up to 200Kbaud reliably). Unless loss of data bytes is a nuisance, this should be of little concern.

- b Abort. Normally an error, but at high clock speeds these can occur on a correctly functioning network (the b occurs in place of the v). One can also get packets like:

C800010080b i
99

where it would appear that an idle has occurred in the middle of a packet! This is due to the receive FIFO register in the SDLC chip, which buffers the data bytes but not the statuses. In fact in real time the idle occurred after the data bytes.

- d Clock missing.
Lots of **d**'s indicates that the clock (to the monitor station at least) is intermittent. Suspect Econet lead, or clock connection in the network, or clock itself.

7.8 A Basic Guide to Econet Protocols.

Hex numbers are preceded by &, all other numbers are in decimal.

A standard Econet interchange might look like this:

```
FE00120080v99 1200FEv00 FE001200900301010203000Bv0D 1200FEv00 i
```

which is one of the messages sent when doing a *CAT.

Each of the groups of numbers separated by a space is called a Packet, and a group of packets constituting one of the legal Econet transfer protocols is called a Message. The packet is the basic unit on the Econet. Packets consist of:

One or more flags	(not displayed on monitor)
Any number of data bytes	
2-byte CRC	(only correctness or incorrectness displayed on monitor)
One flag	(not displayed on monitor)

Data bytes within the packet follow with no gaps and there are special encoding techniques which ensure that the flag pattern does not occur within a packet. The packet structure is known as SDLC, and it is constructed and decoded in hardware by a chip in the Econet circuit. On the BBC machine an MC68B54 is used (IC 89) and on Z80-based machines a Z80 SIO.

You will notice that each packet has the same 4 bytes on the front but the order changed. All Econet packets have a 4-byte header describing where the message is going, and who sent it. Both these 'addresses' are 2-byte quantities, the first byte being a station number and the second a gateway number. Normally the gateway number is zero, unless you have a Bridge on your network in which case this may be non-zero. A zero gateway number addresses your 'local' network.

The first 'address' is the destination address. This is on the front of a packet so that any machine (all machines listen all the time) can tell whether that packet is directed at it or at another machine. The second address is the source address, ie the network address of the machine which transmitted the packet. In the first packet **FE00** was the destination address and **1200** was the source address.

You should see now that, in the example above, 2 packets were going from station &12 to station &FE, and two from station &FE to station &12.

Let us now examine the message in more detail, packet by packet.

The first packet is a scout packet, sent to station &FE (=254, so probably the FS) from station &12 (a client). In addition to the packet header, there are two bytes; the first is a control byte, the second the port number identifying the rest of the message. (NB the word port here is nothing to do with hardware ports, although it has much the same function as an identifier.) The control byte isn't very important as regards the FS interface. it can be used for sequencing and is used extensively for printing.

The port number is much more interesting. Stations which are set up to receive messages can selectively allow messages from <all stations or one particular station> and on <all ports or a specified port>. The port number identifies the data to the receiving station as 'this is the data I wish to SAVE' or 'here are some bytes to be printed' or 'this is a print status enquiry': &99 has identified the message as an FS command.

The second packet is an acknowledgement. The first packet was sent from the client to the FS. The acknowledgement packet goes the other way (notice the header bytes are the other way around) and it is a packet telling station &12 that station &FE has recognised the scout package and is prepared to receive some data.

The third packet is a data packet. After the header there follow data bytes. The protocol does not specify in advance how many bytes are going to be sent, and it is up to the transmitter (station &12) to decide how many he is going to send. If too many are sent an error will be reported at both ends. Here are the data bytes again (less the packet header):

```
90 03 01 01 02      03 00 0B 0D
```

To interpret these bytes we first have to remember the port number in the scout packet, which was &99, the FS command port. This defines the first 5 bytes of data to be a Standard Tx Header (See page 63 of EAUG, page 99 of ESUG), so that the reply port is &90, Function code is 3 (an 'Examine' call, used by *CAT) and context handles are 01, 01 and 02.

The rest of the data bytes are parameters to the examine call (p 65 of EAUG, p.105 of ESUG), which gives ARG=3, and requests &0B entries starting from entry 0 in directory "" ie the CSD). An ARG of 3 tells the FS that the data returned should be file title + access, in ASCII.

The fourth packet is also an acknowledgement. It tells stations &12 that stations &FE has received the data correctly, and that not too much data was sent.

We have looked at a successful Econet transfer. It is also wise to know about transfers that didn't work; the most likely of which is when the error reported is **Not listening**. This can happen in various ways; either the distant machine is not present, or switched off, or it has not been set up for receive or it may have crashed. In either case the monitor output will look like this:

```
FE00120080v99  i
FE00120080v99  i
FE00120080v99  i
FE00120080v99  i
FE00120080v99  i
FE00120080v99  i
```

ie station &12 repeatedly sending scout packets to station &FE but getting no acknowledgement (It sends a few hundred such packets before reporting an error).

This sort of monitor output may also occur under perfectly normal circumstances: if an FS is busy with another client and/or there is disc activity going on it will not be set up to receive packets on its command port, and so will not send any acknowledge packets.

A less likely form of error may look like this:

```
FE00120082vD1 1200FEv00 FE00120000D0A03v31  i
```

(again many times). Station &FE has acknowledged the scout packet from station &12, but has not acknowledged the data packet. Almost certainly too many bytes were sent (the BBC machine will report **Net error**)

8. Econet[®] Machine Code Reference Section

This chapter is still in preparation. For the time being, please refer to the Econet[®] User Guide published by Acorn Computers Ltd.

This chapter will be sent to you in due course.

