Chapter 3:
Utility Programs Reference

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:

3.1 Introduction

Some of the commands described in section 3.3 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.

3.1.1 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 3.0 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>
```

*\<command> Forces the command through to the file server as one of its internal commands eg. **DELETE,RENAME and PRINTER** etc. Note that *CAT and *EX are not commands internal to the file server.

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.

3.1.2 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; those allowed are alphanumerics, 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**.

& 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.

3.1.3 Wild Card Specifiers

A "wild card" in a file specifier allows reference to a group of files. The following three 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 characters, 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 the 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 to be carried out 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.

3.1.4 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.

3.2 Summary of Commands

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 column **Type** describes which machine holds the program that processes the command, in the case of **File Server command** the File Server executes the command internally and no program is loaded in the BBC computer. If the type is **Transient program** then the program will be loaded into the BBC Computer's RAM at page 9 or page E (see BBC Microcomputer User Guide section 40) and will probably not interfere with any existing program in memory. The types **N.F.S. command** and **A.N.F.S. command** are commands that are processed in the BBC microcomputer but do use workspace in page 9 and page E.

The commands described are as follows:

Command	Type	Syntax
*ACCESS	File Server command	*ACCESS <general specifier> <access string>
*ACCOUNT	File Server command	*ACCOUNT <file specifier>[<main account no>] [<(aux account no.>)]
BPUT#	BASIC keyword	See BBC Microcomputer User Guide (Section 33)
BGET#	BASIC keyword	See BBC Microcomputer User Guide (Section 33)
PTR#	BASIC keyword	See BBC Microcomputer User Guide (Section 33)
EXT#	BASIC keyword	See BBC Microcomputer User Guide (Section 33)
EOF#	BASIC keyword	See BBC Microcomputer User Guide (Section 33)
INPUT#	BASIC keyword	See BBC Microcomputer User Guide (Section 33)
PRINT#	BASIC keyword	See BBC Microcomputer User Guide (Section 33)
CLOSE#	BASIC keyword	See BBC Microcomputer User Guide (Section 33)
*BUILD	Transient program	*BUILD <file specifier>
*BYE	N.F.S. command	*BYE
*CAT	BBC Micro O.S. command	*CAT [<directory specifier>] | *. [<directory specifier>]
*CATALL	Transient program	*CATALL [<directory specifier>]
*CDIR	File Server command	*CDIR [<directory specifier>]
*CLOSE	Transient program	*CLOSE
COPIER	BASIC program	CHAIN "COPIER"
*CV	Transient program	*CV
*DEFACCESS	File Server command	*DEFACCESS [<directory specifier>] <access string>
*DELETE	File Server command	*DELETE <general specifier>
*DIR	File Server command	*DISCS
*DISABLE	File Server command	*DISABLE | *DISABLE SAVES | *DISABLE LIBRARY
*DISCS	Transient program	*DISCS
*DUMP	Transient program	*DUMP <file specifier> [<offset>]
*ENABLE	File Server command	*ENABLE | *ENABLE SAVES | *ENABLE LIBRARY
ERAQ	BASIC program	CHAIN "ERAQ"
*EX	N.F.S. command	*EX [<directory specifier>]
*EXEC	BBC Micro O.S. command	*EXEC <file specifier>
*FREE	Transient program	*FLUSH
*FLUSH	File Server command	*FREE
*FS	A.N.F.S. command	*FS [<network number>.] <station number>
*FSLIST	Transient program	*FSLIST
*GNET	Transient program	*GNET
*GO	Transient program	*GO <32 bit address>
*GTIME	Transient program	*GTIME
*GUSER	Transient program	*GUSER
*HELP	BBC Micro O.S. command	*HELP
*I AM	N.F.S. command	*I AM [<network number>.][<File Server number>] <User ID> [<password>]

*INFO	File Server command	*INFO [<general specifier>]
*LIB	File Server command	*LIB <directory specifier>
LOAD	BASIC command	LOAD "<file specifier>" |
		LOAD <string variable>
*LOAD	BBC Micro O.S. command	*LOAD <file specifier> [<load address>]
*LOGON	Transient program	*LOGON
MULTICOPY	BASIC program	CHAIN "MULTICOPY"
*NOTIFY	Transient program	*NOTIFY <station number> |
		*NOTIFY <User Id> <text>
OPENIN	BASIC keyword	<numeric variable>=OPENIN "<file specifier>" |
		<numeric variable >=OPENIN <string variable>
OPENUP	BASIC 2 keyword	<numeric variable>=OPENUP "<file specifier>" |
		<numeric variable >=OPENUP <string variable>
OPENOUT	BASIC keyword	<numeric variable>=OPENOUT "<file specifier>" |
		<numeric variable >=OPENOUT <string variable>
*OPT1	BBC Micro O.S. command	*OPT1,<number>
*OPT4	BBC Micro O.S. command	*OPT4,<number>
OSCLI	BASIC 2 keyword	OSCLI "<string>" |
		OSCLI <string variable>
*PASS	File Server command	*PASS <old password> <new passwrod>
*PATHNAME	Transient program	*PATHNAME
*PRINT	Transient program	*PRINT
*PRINTER	File Server command	*PRINTER |
		*PRINTER <logical printer name>
*PRINTOUT	File Server command	*PRINTER <file specifier>
*PROT	Transient program	*PROT
*PROTEX	Transient program	*PROTEX
*PS	Transient program	*PS <station number> |
		*PS |
		*PS <logical printer name>
*PSLIST	Transient program	*PSLIST
*PTIME	Transient program	*PTIME
*PDATE	Transient program	*PDATE
*PDATE2	Transient program	*PDATE2
*PUSER	Transient program	*PUSER
*PUTGET	Machine code program	*PUTGET
*PUTGET2	Machine code program	*PUTGET2
*REMOTE	Transient program	*REMOTE <station number > |
		*REMOTE <User Id.>
*RENAME	File Server command	*RENAME <old general specifier> <new general specifier>
*REROUTE	File Server command	*REROUTE <print job name> <logical printer name>
*ROFF	N.F.S. command	*ROFF
*RUN	BBC Micro O.S. command	*RUN <file specifier > |
		*/<file specifier> |
		*<file specifier>
SAVE	BASIC command	SAVE "<file specifier>" |
		SAVE <string variable>
*SAVE	BBC Micro O.S. command	See command details
*SDISC	File Server command	*SDISC <disc name>
*SPOOL	BBC Micro O.S. command	*SPOOL <file specifier> |
		*SPOOL
*STATEMENT	Transient program	*STATEMENT
*STATIONS	Transient program	*STATIONS [<network number >]
*TIME	Transient program	*TIME
*TYPE	Transient program	*TYPE <file specifier>
*UNPROT	Transient program	*UNPROT
*USERS	Transient program	*USERS
*VERS	Transient program	*VERS
*VIEW	Transient program	*VIEW <station number> | <user identifier>

Syntax: *ACCESS ||B<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 (see this Section).

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.

Access letters which apply to the current user will be given in capitals, and thoe others in lower case.

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 saved over, loaded, or opened for reading using OPENIN (see this Section).

WR Read and write access allowed: this means that the file can be loaded, or opened for update using OPENUP (see this Section)

W Write only access: this means 'append to end only'. The file can 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,saved over or renamed until access is changed. Users with *public access* can never delete or rename a file, so this applies only to *owners*. 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).

/spl Item is waiting to be printed. This access code is given automatically to entries spooled to the print queue directory, %PRINTQ, and cannot be changed. *Owners* have read access to these entries and *non-owners* have none.

/prt Item is waiting to be printed. This access code is given automatically to entries generated in %PRINTQ by the ***PRINTOUT** command (see this Section), and cannot be changed. These

entries contained pointers to the file to be printed out; *owners* have read access only to this information if they wish.

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 user's 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 INAF	error INAF
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		
Effective Access	BGET#	BPUT# Read EOF	Set/Read PTR#,
D	error INA	error INAF	error INAF
W	error IA unless PTR# = EXT#	error IA	OK
R	OK	error NOFU	OK
WR	OK	OK	OK

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

Error IA is the error message "Insufficient access" (Error BD)
Error INAF is the error message "xxx is not a file" (Error B5)
Error NOFU is the error message "File not open for update" (Error C1)

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 **Error 168 (&A8)**
An attempt to set an illegal attribute (e.g. W to a directory or D to a file).

Insufficient Access **Error 189 (&BD)**
Only an *owner* can change the access status of a file.

Bad attribute **Error 207 (&CF)**
Attempt to use letters other than P D W L R.

Associated Keywords:

*DEFACCESS

Compatibility Notes:

This command is supported by Acorn systems, but access letters **P** and **/spl** are not, nor are the '+' and '-' characters. Note also that Acorn systems define file ownership differently, and that there is no ***DEFACCESS** 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 three digit hexadecimal numbers, ranging between 0 and 3FF.

When a file or directory 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 item is debited from the balance in the main account. As explained in Section 2.4, a user has owner access to an item 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 item from the 'new' account, and credits the same amount to the 'old' account.

The user must be an owner of the item, 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 items beginning with the letters NEW to account number 25, and auxiliary account 37 (as above).

Likely Errors:

Insufficient access **Error 189 (BD)**
Only an owner can change the account of a file.

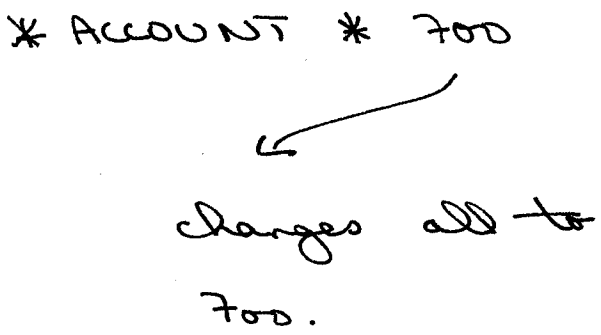
Account nn bankrupt **Error 198 (C6)**
There must be sufficient credit in the new main account.

Associated Keywords:

*CREDIT, *DEBIT, *STATEMENT

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#

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 (see this Section).

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 OSGBPB 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 OSGBPB routine, especially if they find the restrictions on the use of PUTGET make it unsuitable:

Procedure to call OSGBPB from BASIC

At the head of the program allocate workspace,

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

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 §10)

Likely Errors:

xxxx is not a file **Error 181 (B5)**
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.

Insufficient Access **Error 189 (BD)**
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.

File not open for update **Error 193 (C1)**
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#.

Channel **Error 222 (DE)**
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.

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

Compatibility Notes:

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

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 OSGBPB. 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 (see this Section).

Compatibility Notes:

Supported on Acorn systems.

Syntax: *BYE

Description:

This command 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 their station 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.

If **<Ctrl B>** is active (as you are printing) ***BYE** will not close the printing job. **<Ctrl C>** must be used.

Likely Errors:

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

Associated Keywords:

*LOGOFF

Compatibility Notes:

Supported by Acorn systems.

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

Action with Wild Cards in the Directory 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 numbers 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 (see this Section). 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, in alphabetical order, 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         WR/r            File23        WR/r
  ...
  ...

*. PROGS
```

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

```
PROGS (038)                 Owner
MAIN-1                      Option 03 (Exec)
Dir. John                   Lib. LIBRARY

ADDER       WR/r            ScreenDump WR/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:

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

xxxx not found **Error 214 (D6)**
If the directory specified cannot be found.

Associated Keywords:

*CATALL, *EX, *INFO, SIZER

Compatibility Notes:

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

Syntax: *CATALL [<directory specifier>]

Action with Wild Cards in the Directory Name:

Occurs on first match only (alphabetically).

Description:

*CATALL produces a catalogue of the specified directory, and of all sub-directories within it. If the specifier is omitted, this catalogue is given for the currently selected directory. 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 specified directory. 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.

Examples:

```
*CATALL

    !BOOT
    !MAIL
    BTWiring
    contnts
    Ecommands

Directory Letters
         ADDRESSES
         Base

         Directory Bloggs
             12-7-85
             13-5-85
         cougar3

         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:

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

xxxx not found **Error 214 (D6)**
If the specified directory cannot be found.

Associated Keywords:

*CAT, *EX, *INFO, SIZER

Compatibility Notes:

Supported by Acorn systems.

Syntax: *CDIR <directory specifier>

Action with Wild Cards in the Directory 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 main or 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:

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

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

Insufficient access **Error 189 (BD)**
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.

xxxx is not a directory **Error 190 (BE)**
If a file (not a directory) of that name already exists.

xxxx not found **Error 214 (D6)**
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.

Associated Keywords:

*ACCOUNT, *DEFACCESS

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).

Syntax: *CLOSE

Description:

This program closes all currently open files, but not any open directories. 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.

Associated Keywords:

*BYE

Compatibility Notes:

Supported by Acorn systems.

Syntax: `CHAIN "COPIER"`

Action with Wild Cards in the File Name:

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, enter the names of the two systems (e.g. DISC and NET) after the appropriate prompts. To copy between two directories, enter ***DIR <directory pathname>** after each, and similarly to copy between File Servers type ***I AM <File Server station number><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. The system will then contiue to prompt for files to be copied until **<Escape>** is pressed.

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 (see this Section) 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.

Associated Keywords:

*MULTICOPY

Compatibility Notes:

Supported by Acorn systems.

*COPY

Syntax: *COPY <source file specifier> <dest. file specifier>

Description:

This program makes a copy of a file. Any user program or text in the computer is not corrupted as locations &900-&AFF are used as the buffer for transferring the data.

Examples:

```
*COPY letter &.text.letter10
```

Likely Errors:

Errors produced by *LOAD, *SAVE, OPENIN and OPENOUT can occur when running this program.

There are no other errors specific to this program.

Associated Keywords:

COPIER, MULTICOPY

Compatibility Notes:

Supported by Acorn systems.

RISC OS Notes:

RISC OS command.

Syntax: *CV

Description:

This program displays the station numbers of the currently selected File Server and Printer Server, the user's own station number, and the user id used to log on to the File Server. If there are multiple networks joined by bridges, then the network number will be returned. If the Printer Server is an SJ Research File Server then the name of the currently selected printer will be displayed next to the Printer Server station number.

If your computer has an old NFS rom such as NFS3.34 then a warning to this effect will also be displayed. Old NFS roms have some serious bugs and should be replaced with NFS3.60 or DNFS. An image of NFS 3.60 is supplied in the $.LIBRARY on the release disc and you are free to erase the old NFS rom and replace it with the NFS3.60 image.

Examples:

```
*CV

FS number 254
PS number 254 LASER
You are 001.005
User Id FRED
```

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.

Associated Keywords:

*FSLIST, *PSLIST

Compatibility Notes:

Supported by Acorn systems.

RISC OS Notes:

RISC OS version supplied in $.ArthurLib.

Syntax: *CV

Description:

This program displays the station numbers of the currently selected File Server and Printer Server, the user's own station number, and the version of OSARGS in use. If there are multiple networks joined by bridges, then the network number wile 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

FS number 254
PS number 235
You are 001.005
OSARGS ver 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.

Associated Keywords:

*FSLIST, *PSLIST

Compatibility Notes:

Supported by Acorn systems.

Syntax: *DEFACCESS [<directory specifer>] <access string>

Action with Wild Cards in the Directory Name:

Occurs on first match only (alphabetically).

Description:

This command applies to the directory specified, or if the specifier is omitted to the CSD. The string will be applied as the default access string to any new files created within this directory, or with any subsequently new files created in 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 (see this Section).

The user must be an owner of the specified directory.

Examples:

```
*DEFACCESS WR/R
```

This sets the default access of the files in the currently selected directory to read and write access for owners and read access only for non-owners. 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 $.SECRET PWR/
```

will cause subsequently created files in the directory $.SECRET to have access letters **PWR/** and subsequently created sub-directories to have access **PD/**.

Likely Errors:

FS Unusual Error 70 (46)
An attempt has been made to set an illegal attribute (e.g. D as part of the default access).

Insufficient access **Error 189 (BD)**
Only an owner can change the default access status of a directory.

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

Bad attribute **Error 207 (CF)**
Attempt to use letters other than P D W L R.

xxxx not found **Error 214 (D6)**
If the directory specified cannot be found. If, in the second example, the directory $.SECRET did not exist, the error **SECRET not found** would occur.

Associated Keywords:

*ACCOUNT

Compatibility Notes:

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

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. Note *\DELETE must be used on ANFS.

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* unlocked 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 (see this Section) 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 beginnning 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:

Directory not empty **Error 180(B4)**
Caused by an attempt to delete a directory that still contains some entries.

Insufficient access **Error 189 (BD)**
Caused if the user is not an owner (i.e. does not have access to the main or auxiliary account) of the item.

Not ENABLED **Error 189 (BD)**
If the system manager has set the ***ENABLE required** option for this user, the **Not ENABLEed** error will be given.

Already opened by xxxx **Error 194 (C2)**
If another user has this file open for reading or writing, then it cannot be deleted until it has been closed.

Syntax: *DESCRIBE [<topic>]

Description:

This program displays help on various system utilities and also on error messages. The help is hierarchical so *DESCRIBEing a topic may reveal further sub-topics on which help is available.

The program works by *TYPEing text files in a **!help** sub-directory of the Library. It is possible to add your own help files to the system by preparing the text in a word processor and then spooling it out into the appropriate place in the !help directory structure.

Examples:

```
*DESCRIBE
  Errors
  Utilities

*DESCRIBE Utilities.logon

Syntax
======
*LOGON

Description
===========
A secure method of logging on to the
file server as your password is not
displayed.

Examples
========

*LOGON

User Id :ARBS
Password:******
```

Likely Errors:

Errors produced by *LOAD, *SAVE and OPENIN can occur when running this program.

There are no other errors specific to this program.

Associated Keywords:

*HELP

Compatibility Notes:

Supported by Acorn systems.

RISC OS Notes:

RISC OS version supplied in $.ArthurLib. The RISC OS version access the same text files as the BBC version ie files in $.Library.!help on drive 0.

Entry locked **Error 195 (C4)**
It will be necessary to use the *ACCESS command to unlock the file before it can be deleted.

Bad wildcard **Error 204 (CC)**
Use of a wildcard on an ANFS without using *\DELETE

Associated Keywords:

*ENABLE, ERAQ

Compatibility Notes:

Supported by Acorn systems, but note that wild cards are not allowed.

Syntax: *DIR [<directory specifier>]

Action with Wild Cards in the Directory 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.

There are several special characters that can be used to specify directories: & can used to represent the URD in pathnames, ^ refers to one level up the directory tree each time it is used, and @ specified the CSD. The last option is useful if a program asks for a directory name to add to *DIR, as simply pressing <Return> would go back to your URD.

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 directory two levels up the tree from your URD as the new CSD.

Likely Errors:

xxxx not found **Error 214 (D6)**
If the specified directory could not be found.

Too many files open **Error 192 (C0)**
If several files are open for random access, it is possible that there may not be a channel left for the CSD.

xxxx is not a directory **Error 190 (BE)**
Caused by an attempt to specify a file as the CSD.

Associated Keywords:

*DISC

Compatibility Notes:

Supported by Acorn systems, but wild cards and the character ^ are not allowed.

Syntax: *DISABLE || *DISABLE SAVES || *DISABLE LIBRARY

Description:

This command is the exact opposite of the *ENABLE command (see 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 by the system manager in the password file.

***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 SJ Research 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 the that of *DISABLE (without an argument).

Associated Keywords:

*ENABLE

Compatibility Notes:

Not supported by Acorn systems.

Syntax: *DISCS

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. It will not list the tape drive; although this, if fitted, is treated as a virtual disc called %TAPE on the File Server.

This command is identical to *FREE (see this Section), 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,   used

   0   DISC1        560k         481k
   1   DISC55       1098k        113k
```

Likely Errors:

There are no errors specific to this program.

Associated Keywords:

*FREE

Compatibility Notes:

Supported by Acorn systems.

Syntax: *DUMP <file specifier> [<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 nn  cccccccc
```

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 Sections 5.5 and 6.5 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.

Associated Keywords:

*TYPE

Compatibility Notes:

Supported by Acorn systems.

Syntax: *ENABLE || *ENABLE SAVES || *ENABLE LIBRARY

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** (see this Section) 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 the error BD will occur, and the message **Not ENABLEd** will be given.

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 reason for not allowing files of shorter than 16 bytes to be 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 Research File Servers (but not for Acorn FSs), but it may have been turned off with ***DISABLE LIBRARY**.

The effect of these *ENABLE commands persists until the user logs off, or until he types the correspondinG *DISABLE (see 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.

Associated Keywords:

*DISABLE

Compatibility Notes:

Not supported by Acorn systems.

Syntax: CHAIN "ERAQ"

Action with Wild Cards in the Directory Name:

Occurs on first match (alphabetically).

Description:

This program deletes part or all of an entire directory tree.

The first question will be:

```
Do you want 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). When it stops, the CSD will be changed to the last directory visited by the program.

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.

Associated Keywords:

*DELETE

Compatibility Notes:

Supported by Acorn systems, but can be used only by a system privileged user.

Syntax: *EX [<directory specifier>]

Action with Wild Cards in the Directory 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

<job name> <User Id> at Stn. sss <length> <access> <printer> <date1><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.
<User Id>	is the user who submitted the job for printing.
<printer>	is the logical printer selected for the job.
xx	is the account number associated with the file (see under *ACCOUNT).
yy	is the auxiliary account number associated with the file.
sss	is the station number from which the print job was submitted.
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 F0 (00)
GT-Eg             00000800        00008023   00017C   WR/r       11jul85
15jul85 14:30 F0 (81)
PUTGET            00000000        FFFFFFFF   0013B5   WR/r       08jun85
11jun85 18:55 F0 (00)
temp              00000000        FFFF3200   002231   WR/r       15jul85
15jul85 14:28 F0 (00)
wombat            FFFFFFFF        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:

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

xxxx Not Found **Error 214 (D6)**
If the directory specified cannot be found.

Associated Keywords:

*CAT, *CATALL, *INFO, SIZER

Compatibility Notes:

Similar on Acorn systems, but without account information, or the date of original creation field. Print spooling is also not avaliable on Acorn systems. A number known as the SIN will appear in the space in which an SJ Research File Server would display the account numbers.

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 from 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 utilty, or the *SPOOL command (see this Section), 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 (see this Section) 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:

File not found **Error 214 (D6)**
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 (see this Section) and so can cause the same errors as that command.

Associated Keywords:

*BUILD

Compatibility Notes::

Supported by Acorn systems.

Syntax: *FLUSH [<job name>]

Action with Wild Cards in Job Name:

Occurs on every match

Description:

This command causes printout to be flushed. It will be found useful if a user's program has generated large quantities of spurious output.

When a user issues this command without a parameter all printout sent by the user from that station will be cleared. This could include the job the file server is currently outputting to the printer.

To selectively remove files from the print queue *FLUSH should be used with a job name. This name can include wildcards for deleting more than one file at a time.

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 a printer's internal buffer, it will be necessary to turn the printer off and on.

Likely Errors:

There are no errors specific to this command

Associated Keywords:

*MFLUSH, *PGO, *PSTOP

Compatibility Notes:

Not supported by Acorn systems.

RISC OS Notes:

Compatible with RISC OS.

Syntax: *FLUSH

Description:

This command causes the contents of any printout to be flushed. It will be found useful if a user's program has generated large quantities of spurious output.

Any printout, waiting to be printed, in the %PRINTQ directory belonging to the user will also be cleared. To determine whether the files in the print queue belong to the user typing *FLUSH, the station number of the computer and the user identifier must be the same as the user issuing the command.

To selectively remove files from the print queue *DELETE should be used.

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 a printer's internal buffer, it will be necessary to turn the printer off and on.

Likely Errors:

There are no errors specific to this command

Associated Keywords:

*PGO, *PSTOP

Compatibility Notes:

Not supported by Acorn systems.

Syntax: *FREE

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. If your File Server is fitted with a tape drive, this will be treated as a virtual disc called %TAPE, but will not be listed by *FREE.

Examples:

```
*FREE
```

will produce the following response from the system (for example):

```
Drive  Name        Bytes free   Bytes used

   0   DISC1         522k         21086k
   1   Main-05      3496k         18112k
```

Likely Errors:

There are no errors specific to this program.

Associated Keywords:

*DISCS

Compatibility Notes:

Supported by Acorn systems.

Syntax: *FS [<network number>.]<station number>

Description:

*FS is only available on the Advanced version of the N.F.S. ROM (ANFS); see under ***HELP** for how to find out which ROMs are fitted to your station.

This command allows you to change the File Server station number stored in the BBC Microcomputer, while it stores the handles you were using on the File Server you have just left. Handles can be stored for up to five File Servers at once, although this depends on how many files you have open.

Thus you can swop between File Servers which you are logged on to, without having to repeatedly log on and lose your CSD and open files.

Likely Errors:

If a station which is not a File Server is selected, the message **Station nn not listening** will be given when any filing operations are attempted.

Associated Keywords:

*FSLIST, *I AM

Compatibility Notes:

Supported by Acorn systems for stations with the ANFS ROM.

Syntax: *FSLIST

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 File Server ver M.97/HDFS
064.127  SJ Research File Server ver 0.91/FDFS
064.182  SJ Research File Server ver 0.90/FDFS
064.235  SJ Research File Server ver M.98/MDFS
    200  SJ Research File Server ver M.97/HDFS
```

Likely Errors:

There are no errors specific to this command.

Associated Keywords:

*FS

Compatibility Notes:

Supported by Acorn systems.

Syntax: *GNET followed immediately by a GET statement.

Description:

This command puts a character into the BBC Microcomputer's buffer, so that it can be read by a program using a GET command (which must follow *GNET immediately). The value read will be the network number for your station, which will be in the range 0 to 255. This number is required if the network being used contains *bridges* to other networks, in order to access a station on the other side of a bridge.

Do **not** the use BASIC keyword INPUT with *GNET, since any bytes which are interpreted as control characters will not be stored, 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 bytes will be entered into the keyboard buffer, possibly causing strange effects.

With bridges this information is also useful when you wish to log-on to a particular File Server, and you do not know whether or not it is on the local network for your particular station (See ***I AM**). If the File Server is at station 250 on network 3, you will need to log-on with:

```
*I AM 0.250 DIANA | *I AM 250 DIANA
```

if your station is on network 3, the local network for the File Server. However if your station is on network 2, it is necessary to specify the full station number of the File Server, including the network number, by typing:

```
*I AM 3.250 DIANA
```

The full station number of a File Server on your local network will not permit you to log-on and will give the message **Not Listening**, as if the File Server were not present. The File Server must now be specified as 0.250 as the default network has been changed to the inaccessible network 3 by the previous command.

Examples:

```
10 *GNET
20 net%=GET

30 IF net%=1 THEN room$="office"
40 IF net%=64 THEN room$="coffee room"

50 PRINT "You are now in the ";room$
```

Likely Errors:

Note that this command will not work on stations fitted with the N.F.S. ROM version 3.34. Also the BBC Microcomputer screen may do strange things if a mistake occurs between executing *GNET and GETting the byte.

Compatibility Notes:

Supported by Acorn systems with appropriate ROMs.

Syntax: *GO <32 bit address>

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 first 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 if 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 is fitted.

```
*/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.

*GOODBYE

Syntax: *GOOODBYE

Description:

This command logs a user off all File Servers on a network. Open print jobs will also be closed. A message confirming successful logoff is displayed for each File Server.

Examples:

```
*GOODBYE
    251 Logged off OK
065.019 Logged off OK
```

Likely Errors:

There are no errors specific to this command.

Compatibility Notes:

Supported by Acorn systems however Acorn File Servers will be displayed in the list of the File Servers logged of from several times, even if you were never logged on in the first place.

RISC OS Notes:

Use the RISC OS *SHUTDOWN command instead.

Syntax: *GTIME followed immediately by 5 GET statements.

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 &0F = month (between 1 and 12)
(second byte AND &F0) 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 stored, 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.

Associated Keywords:

SETIME

Compatibility Notes:

Supported by Acorn systems that contain a real-time clock (otherwise the time may be nonsense).

Syntax: *GUSER
INPUT ""user$

Description:

This program plants the name of the user logged on at the current station into the BBC Microcomputer, so that it can be read into a string by the BASIC keyword INPUT -- this must follow *GUSER immediately. The double quote characters mean that a null string is used as a prompt for the INPUT statement, instead of the usual ?.

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.

Associated Keywords:

*I AM, *PUSER, *CV, *USERS

Compatibility Notes:

Supported by Acorn systems, but note that some Acorn File Servers support user names of more than 10 characters.

Syntax: *HELP

Description:

This command will list all the ROMs in the BBC Microcomputer which respond to it. This is particularly useful for finding out which version of the N.F.S. ROM your station is fitted with, as this affects some network operations. *HELP may also be used with a parameter to give further information about some ROMs.

Examples:

```
*HELP

6502 TUBE 1.10

NFS 3.60

OS 1.20
```

Likely Errors:

There are no errors specific to this command.

Compatibility Notes:

Supported by Acorn systems.

Syntax:

***I AM** [<network number>.][<File Server number>] <User Id> [<password>]

***I AM** [<network number>.][<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 the directory specified as the User Root Directory (URD) for the user in the password file, usually with the same name as the user, and will select this as his Currently Selected Directory (CSD) -- see Section 2.3.

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 an 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 (see this Section) has a similar effect, but works for all versions of N.F.S. ROM. 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 automatically 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 is certainly vital if security is important.

Note that it is possible for one user to be logged on to a File Server from several stations; and for a user at one station to be logged on to several File Severs from the File Servers' point of view. This can be useful for printing, and the command ***FS** (see this Section) may be used to switch a station between such File Servers.

Examples:

```
*I AM JOHN
```

logs on to the previously specified File Server (or the File Server at station 254 if none was 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:

Not listening **Error 162 (A2)**
Either there exists no File Server at the specified station number, or it has been switched off.

No clock **Error 163 (A3)**
The user's station is not plugged into the network, or the network clock is not running. See Section 9.4.

Bad password **Error 185 (B9)**
An attempt to use illegal characters in the password field (e.g. * # $ %) will give this error.

Wrong password **Error 187 (BB)**
If the user makes an error in entering his password.

User not known **Error 188 (BC)**
If there is no default user, this error will be given if the user's name is not known to the system.

File not found **Error 214 (D6)**
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.

Associated Keywords:

```
*BYE, *FS, *LOGON, *LOGOFF, *PASS
```

Compatibility Notes:

Supported on Acorn systems.

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

<job name> <User Id> at Stn. sss <length> <access> <printer> <date1><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.
<User Id>	is the user who submitted the job for printing.
<printer>	is the logical printer selected for the job.
<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.
sss	is the station number from which the print job was submitted.
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 (see 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 $.FRED
```

will give informtion on $.FRED, which is probably a user root directory for user FRED.

```
*INFO
```

gives information on the currently selected directory.

Likely Errors:

Not found **Error D6 (214)**
If the specified item is not found.

Associated Keywords:

*CATALL, *EX, *INFO, SIZER

Compatibility Notes:

Supported by Acorn systems.

Syntax: *LIB <directory specifier>

Action with Wild Cards in the Directory 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 (see this Section) will automatically select the library directory selected in the password file for the user 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. The selected library directory defaults to $.LIBRARY on logging-on if no particular library was specified in EDITPASS.

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:

xxxx is not a directory **Error 190 (BE)**
If a file name has been specified

xxxx Not Found **Error 214 (D6)**
If the directory specified does not exist.

Associated Keywords:

*ENABLE

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.

Syntax: LOAD "<file specifier>" | LOAD <string variable>

where the <string variable> must contain a legal file specifier.

Action with Wild Cards in the 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:

xxxx is not a file **Error 181 (B5)**
You cannot LOAD a directory.

Insufficient access **Error189 (BD)**
There must be access status **R** for this user, otherwise he will not be able to load the file.

Bad name **Error 204 (CC)**
If the file name contains illegal characters (e.g. **$ % . ^:** except in contexts where they are permitted).

xxxx Not Found **Error 214 (D6)**
If the specified file did not exist.

Associated Keywords:

SAVE, *ENABLE, *DISABLE

Compatibility Notes:

Supported by Acorn systems. Note that Acorn File Servers do *not perform a library search with the LOAD command.* The command ***DISABLE LIBRARY** sets the library search, on an SJ Research File Server, to be the same as the Acorn File Server.

Syntax: *LOAD <file specifier> [<load address>]

Action with Wild Cards in the 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 (see this Section).

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:

xxxx is not a file **Error (181) B5**
You cannot *LOAD a directory.

Insufficient access **Error189 (BD)**
There must be access status **R** for this user, otherwise he will not be able to load the file.

Bad name **Error 204 (CC)**
If the file name contains illegal characters (e.g. **$ % . ^:** except in contexts where they are permitted)

xxxx Not Found **Error 214 (D6)**
If the specified file did not exist.

Associated Keywords:

*SAVE

Compatibility Notes:

Supported by Acorn systems. Note that Acorn File Servers do *not perform* a library search with the *LOAD command.

Syntax: *MFLUSH <User Id> [<station number>]

Description:

This command can flush multiple print jobs. It is allows owners of the print queue directory to quickly remove large numbers of print jobs. The print queue directory is scanned for jobs which were printed by the specified user and these are then flushed using the *FLUSH command. If the optional station number is also specified then only those jobs which were printed by the user at that station will be flushed.

When a print job is flushed the full info, as displayed by *INFO, is output on the screen.

Examples:

```
*MFLUSH BOOT
AA67        BOOT        at Stn.132 000123 /spl    hold     today  12:19 03F (000)
AB20        BOOT        at Stn.167 0004FD /spl    parall   today  13:45 03F (000)
BA01        BOOT        at 034.100 001E34 /spl    hold     today  16:01 03F (000)

*MFLUSH BOOT 132
AA67        BOOT        at Stn.132 000123 /spl    hold     today  12:19 03F (000)
```

Likely Errors:

There are no errors specific to this command

Associated Keywords:

*FLUSH, *PGO, *PSTOP

Compatibility Notes:

Not supported by Acorn systems.

RISC OS Notes:

RISC OS version supplied in $.ArthurLib.

Syntax: *LOGON

Description:

This program prompts a user for user identifier and password, then logs that user 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 station 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.

Associated Keywords:

*I AM, LOGOFF, *FS,*BYE, *PASS

Compatibility Notes:

Supported by Acorn systems.

Syntax: `CHAIN "MULTICOPY"`

Action with Wild Cards in the Directory Name:

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**, then all files and directories will be put in the main account of the destination directory.

There is also an option to copy the creation dates of the files; this is intended for use when backing up the File Server. The system manager may set this option so that ordinary users cannot use it.

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 V1.05

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 (Y/N): Y
Do you wish to copy account information (Y/N): Y
Do you wish to copy creation date etc.
(for system manager's use only) (Y/N): N
source directory name      : PROGS
destination directory Name: PROGS

              .
              .
              .
```

(list of the files being copied)

Syntax: CHAIN "MULTICOPY"

Action with Wild Cards in the Directory Name:

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 force overwriting of locked files (Y/N):** -- if the user answers **Y**, it will overwrite existing files even if they are locked. The next questions asks **Do you wish to include sub-directories (Y/N)** -- 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**, then all files and directories will be put in the main account of the destination directory.

There is also an option to copy the creation dates of the files; this is intended for use when backing up the File Server. The system manager may set this option so that ordinary users cannot use it.

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 V1.11

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
(or press RETURN to use current 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 force overwriting of
locked files (Y/N) :Y
Do you wish to include sub directories  (Y/N) :Y
Do you wish to copy account information (Y/N) :N
Do you wish to copy creation date etc.
(for system manager's use only) (Y/N) :Y
source directory name      :PROGS
destination directory name :PROGS
            .
            .
            .
```

(list of the files being copied)

The next example shows a copy of the directory structure '$.RELEASE' from a hard disc called 'MAIN1' to a floppy disc called 'Main2', on the same File Server.

```
CHAIN "MULTICOPY"
Multiple file copy utility V1.11

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
(or press RETURN to use current FS) :
*I AM 254 FRED
Log-on text for dest.  FS
(or press RETURN for same FS) :
*I AM
Do you wish to force overwriting of
locked files (Y/N) :Y
Do you wish to include sub directories  (Y/N) :Y
Do you wish to copy account information (Y/N) :N
Do you wish to copy creation date etc.
(for system manager's use only) (Y/N) :N
source directory name      :$*1.RELEASE
destination directory name :$*2.NEWREL.ANOTHER
             .
             .
             .
```

(list of the files being copied)

Error Handling:

If an error occurs during a copying operation then MULTICOPY will produce a * prompt at which you can type the commands necessary to fix the problem. In particular it allows errors such as **Already open by**, **Account bankrupt** and **Too short** to be fixed. When all the necessary * commands have been typed pressing just <RETURN> at the * prompt will bring up the prompt **R(etry) or S(kip)** . Selecting R means MULTICOPY will resume trying to copy the file which caused the error. Selecting S means that the file which caused the error will be skipped and copying will resume at the next file. This is important should you encounter a file which cannot be copied due to an error which cannot be fixed, eg a disc error.

Backing Up to MDFS Floppy:

MULTICOPY can be used to backup to MDFS format floppy discs. This is achieved by simply entering the source and destination directories with the relevant disc name prefix. With care it is possible to swap floppy discs duing the copying operation and thus backup more then 800k in one operation. Log on to the File Server using a user name from the password file on the hard disc. Create a set of floppy discs for backup purposes and name them Backup1, Backup2, Backup3 etc. As the source directory specify the hard disc eg **$HARD** and as the destination directory specify the floppy disc using a wildcard which will match all the floppy discs names eg **$Backup***. Eventually the copying operation will produce the **Disc full** error and hence the * prompt. At this moment the program is still accessing the floppy disc so removing the disc would cause a fatal error. Press <RETURN> to bring up the R(etry) or S(kip) prompt. At this moment the program is accessing the hard disc so it is possible to press the **Release Discs** button and swap to the next backup floppy. Press R to retry the file which caused the previous floppy disc to become full and copying will be resumed. All the necessary sub-directories will be created on the new floppy disc.

Likely Errors:

Insufficient access **Error189 (BD)**
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.

xxxx is not a directory **Error 190 (BE)**
If the user has specified a file as the source or destination directory name prompt.

Already opened by xxxx **Error 195 (C2)**
MULTICOPY will save over a file of the same name. If this file was already open, this error will occur.

Locked **Error 195 (C3)**
After an attempt to save over a file of the same name, if the latter was locked.

xxxx Not Found **Error 214 (D6)**
If the source directory could not be found.

Account xxxx bankrupt **Error 198 (C6)**
If the account number being saved to does not have sufficient credit.

Associated Keywords:

COPIER

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.

RISC OS Notes:

Use either the RISC OS *COPY command or the desktop copy facility instead.

Syntax: *NEWPASS

Description:

This command is the allows a user to change their password. It is the most secure method as it makes use of an encryption key to encode both the user's old password and the new one, before sending the change password command to the File Server.

The new and old passwords are not displayed on screen but keypresses are echoed as asterisks. The **New Password** prompt will appear twice and the user must enter the same new password twice. This is to prevent the user from accidentally setting their password to something they do not know as the result of a mistype. If the two entries of the new password are not the same the **Retype PW** error message will be given and the user will need to type *NEWPASS again.

A minimum password length of six characters is recommended for the encryption algorithm to be fully effective.

Examples:

```
*NEWPASS
Old Password: *****
New Password: ********
New Password: ********
```

Likely Errors:

Password file changed **Error 3 (03)**
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.

Bad password **Error 185 (BD)**
There is an illegal character in the password quoted, probably * # $ % ^:

Wrong password **Error 187 (BB)**
The old password does not match the one stored.

Associated Keywords:

*LOGON, *PASS, *I AM

Compatibility Notes:

Not supported by Acorn systems.

RISC OS Notes:

RISC OS version supplied in $.ArthurLib.

The next example shows a copy of the directory structure '$.RELEASE' from a hard disc called 'MAIN1' to a floppy disc called 'Main2'.

```
CHAIN "MULTICOPY"
Multiple file copy utility V1.05

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):
*
Do you wish to include sub-directories (Y/N): Y
Do you wish to copy account information (Y/N): N
Do you wish to copy creation date etc.
(for system manager's use only) (Y/N): N
source directory name      : $*1.RELEASE
destination directory Name: $*2.NEWREL.ANOTHER

                .
                .
                .
```

(list of the files being copied)

Likely Errors:

Insufficient access **Error189 (BD)**
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.

xxxx is not a directory **Error 190 (BE)**
If the user has specified a file as the source or destination directory name prompt.

Already opened by xxxx **Error 195 (C2)**
MULTICOPY will save over a file of the same name. If this file was already open, this error will occur.

Locked **Error 195 (C3) Locked**
After an attempt to save over a file of the same name, if the latter was locked.

xxxx Not Found **Error 214 (D6)**
If the source or destination directories could not be found.

Account xxxx bancrupt **Error 198 (C6)**
If the account number being saved to does not have sufficient credit.

Associated Keywords:

COPIER

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.

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 (see this Section) downward.

Description:

This utility produces the message *| **<sender station number>: <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 (see this Section).

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
```

The message will be accompanied by a <ctrl-G> character, which produces a beep.

Likely Errors:

Net Error **Error 161 (A1)**
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 Chapter 8).

Not listening **Error 162 (A2)**
If the destination station is switched off, or if its protection byte has been set.

Not logged on **Error 174 (AE)**
If a user identifer was specified, and that user was not logged on at the currently selected File Server.

Associated keywords:

*PROT, FORCER

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.

Syntax: <numeric variable>=OPENIN "<file specifier>"
<numeric variable>=OPENIN <string variable>

Action with Wild Cards in the 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 @ 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 the 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 (see this Section). 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:

Insufficient access **Error 189 (BD)**
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.

Too many files open **Error 192 (C0)**
There is a limit to the number of channels available, normally 5 or 6.

File not open for update **Error 193 (C1)**
Will be caused by an attempt to use BPUT# or PRINT# after OPENIN.

Already opened by xxxx **Error 194 (C2)**
It is not permitted to open a file for input, if it already open for output by either this user or another.

Bad name **Error 204 (CC)**
If the file name contains illegal characters (e.g. **$ % . ^:** except in contexts where they are permitted).

Associated keywords:

```
OPENUP, OPENOUT, *CLOSE, BGET, BPUT
```

Compatibility Notes:

Supported on Acorn systems. Note that Acorn File Servers do **not** perform a library search with the OPENIN command.

Syntax: **<numeric variable>=OPENUP "<file specifier>" |**
<numeric variable>=OPENUP <string variable>

Action with Wild Cards in the 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 (see 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 the 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:

xxxx is not a directory **Error 181 (B5)**
You cannot OPENUP a directory.

Insufficient access **Error 189 (BD)**
There must be access status **W** for this user, otherwise he will not be able to open the file for for updating.

Too many files open **Error 192 (C0)**
There is a limit to the number of channels available, normally 5 or 6.

File not open for update **Error 193 (C1)**
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#.

Already opened by xxxx **Error 194 (C2)**
It is not permitted to open a file for update, if it already open for any purpose by either this user or another.

Bad name **Error 204 (CC)**
If the file name contains illegal characters (e.g. **$ % . ^:** except in contexts where they are permitted)

Associated keywords:

OPENUP, OPENOUT, *CLOSE, BPUT, BGET

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.

Syntax: <numeric variable>=OPENOUT "<file specifier>" |
<numeric variable>=OPENOUT <string variable>

Action with Wild Cards in the File Name:

Wild cards prohibited.

Description:

This BASIC command creates a new file with the 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.

OPENOUT 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:

xxxx is not a directory **Error 181 (B5)**
Caused by attempting to create a file with the same name as that of a directory.

Insufficient access **Error 189 (BD)**
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.

Too many files open **Error 192 (C0)**
There is a limit to the number of channels available, normally 5 or 6.

Already opened by xxxx **Error 194 (C2)**
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.

Entry locked **Error 195 (C3)**
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.

Bad name **Error 204 (CC)**
If the file name contains illegal characters (e.g. **$ % . ^:** except in contexts where they are permitted).

Associated keywords:

OPENI, OPENOUT, *CLOSE, BPUT, BGET

Compatibility Notes:

Supported by Acorn systems, but the space allocation works differently.

Syntax: *OPT1,<number>

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:

Bad command **Error 254 (FE)**
If the second number is in excess of 255.

Associated keywords:

LOAD, *LOAD, SAVE,*SAVE

Compatibility Notes:

Supported on Acorn systems. Some versions will not work with the value of <number> greater than 1.

Syntax: *OPT4, <number>

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 Research 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) logs 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 BA (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:

Insufficient privilege **Error 186 (BA)**
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.

Password file changed **Error 3 (03)**
Caused if the system manager has made any change to the password file since the user logged on. The user should log on again.

Associated keywords:

*I AM, *BUILD !BOOT

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

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.

Associated keywords:

```
*I AM, EDITPASS
```

Compatibility Notes:

Supported on Acorn systems.

Syntax: *PASS <old password> <new password>

Action with Wild Cards in the Password:

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 insufficient privilege (error BA) 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:

Password file changed **Error 3 (03)**
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.

Bad password **Error 185 (BD)**
There is an illegal character in the password quoted, probably * # $ % ^:

Wrong password **Error 187 (BB)**
The old password does not match the one stored.

Associated keywords:

Compatibility Notes:

Supported by Acorn systems. Note that the option to 'lock' the password does not exist on Acorn File Servers.

Syntax: *PATHNAME

Description:

This program prints the full path of the currently selected directory. Thus you can check where you are in a directory structure and which disc you are using.

Examples:

```
*PATHNAME
:MAIN-IV.$.FS-manual.Iss023
```

The currently selected directory is on the disc MAIN-IV and is the directory Iss-023 in the directory FS-manual in the disc root directory.

Likely Errors:

There are no errors specific to this command.

Associated keywords:

Compatibility Notes:

Supported by Acorn systems.

Syntax: *PRINT

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. See Sections 5.5 and 6.5 for a complete explanation of printing through the network.

*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 will only work if the File Server you are logged on to is the printer server you wish to use.

*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%
```

This modification may have already been made, if it is needed for the printers on your File Server.

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-C> <Escape>, then try *PRINT again.

Associated keywords:

*PS

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.

Syntax: *PRINTER || *PRINTER <logical printer name>

Action with Wild Cards in the Printer Name:

Wild cards prohibited.

Description:

This command is intended for use with the *PRINT and *PRINTOUT commands (see 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 logical 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.
jammed	the printer is jammed or has run out of paper, ribbon etc. This message is also given if the print queue directory is full.
not authorised	this user is not authorised to use the selected printer.

If *PRINTER is followed by a printer name, then the internal logical printer selection, within the current File Server for the user's station, will be changed to the new name. Error 0A will be produced if the user is not authorised to select the printer specified. See Section 2.5.5 for a description of logical printers.

*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
```

Similarly the command:

```
*PRINTER ML
```

will produce a response of the form:

```
ML busy with station 003 (FRED)
```

Likely Errors:

Not authorised to use printer Error 10 (0A)
If the system manager has restricted the use of this printer to users with access to a certain account.

Associated keywords:

*PS, *PRINTOUT

Compatibility Notes:

Not supported by Acorn systems.

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 logical 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 appropriate printer server banner and end-text characters (as set up for the logical printer 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 Microcomputer 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 Microcomputer(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 Microcomputer(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 Error

Printer busy with staion nnn(xxxx) Error 9 (09)
If the printer is not currently available.

Not authorisedto use printer Error 10 (0A)
The system manager can set up a printer, so that only users with access to a certain account number can use it.

Insufficient access Error 189 (BD)
The user must have read access to the specified file.

Associated keywords:

*FLUSH, *PRINTER, *REROUTE

Compatibility Notes:

Not supported by Acorn systems.

Syntax: *PROT

Description:

This program protects the station at which it is entered against low level operations such as PEEK and POKE from other stations, and also network commands like ***REMOTE** and ***NOTIFY**. Other machines attempting these operations will behave as if the PROTected station were not there. Protection continues until <Ctrl-Break> is pressed or the command ***UNPROT** is given. More information about this call is given in the machine code section, Chapter 10.

If some operations from other stations are required but not others, an extended version of this command is given by ***PROTEX**.

Likely Errors:

There are no errors specific to this command.

Associated keywords:

*PROTEX, *UNPROT

Compatibility Notes:

Supported by Acorn systems.

Syntax: *PROTEX

Description:

This program offers the user the option of protecting his station against various types of operation from other stations in the network. The user will be prompted with text of the form:

```
Protection against: Halt  (Y/N)
```

The options of protection against Halt, Utils, Prc, Jsr, Poke and Peek are offered. Responses other than **Y** or **N** will be ignored, except <Escape> which causes the machine to hang up, and <Break> which leaves the *PROTEX program.

The **Halt** option stops all action in the machine until a **Continue** command is issued and is thus a good thing to be protected against.

Utils protects against fileserver utilities which affect other stations, in particular FORCER, *NOTIFY, *VIEW and *REMOTE.

Proc disallows all remote procedure calls to a station. Only a few programs use these calls (one is *FAST) and they are not usually implemented without the station user's co-operation, so protection against these is not often needed.

Jsr prevents other stations from taking over areas of your station's memory to run a subroutine for one of their programs. This subroutine will need to have been entered into your station by using POKE.

The **Poke** and **Peek** options respectively prevent other stations from writing to or reading from your station's memory directly.

The ***PROT** command has the effect of setting all of these protections on; whereas ***UNPROT** sets them all off.

Likely Errors:

There are no errors specific to this command.

Associated keywords:

*PROT, *UNPROT

Compatibility Notes:

Supported by Acorn systems

Syntax: *PS <station number> | *PS | *PS <logical printer name>

Action with Wild Cards in the Printer Name:

Wild cards prohibited.

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 *PS, or *PRINT, commands 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 <logical printer name>** will search for a printer server with the specified logical printer. A full discussion of printing through the network is given in Sections 2.5 and section 6.

The system manager will allocate specific names to particular printer/banner combinations, and the use of *PS <logical printer name> will search for a combination with that particular name. There can be several logical printers with the same name, but these should be identical. 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 **Not listening Error 162 (A2)** or **No reply Error 165 (A5)** 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-C> <Escape>, then try *PS again.

Associated keywords:

*PRINTER, *PSLIST

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.

Syntax:*PSLIST

Description:

*PSLIST displays a list of all of the available printer servers on the network. For SJ Research printer servers, the logical printers available on each File Server will be listed after the station number.

Logical printers will not be listed if they have been set by the system manager to be non-existent. However, all other logical printers on an available printer server will be listed. Those you are not allowed to use will be prefixed with the x character.

If the Econet installation comprises multiple networks, *PSLIST will also display printer servers on other networks, preceded by their network number (e.g. printer server 235 on network 2 will be displayed as 002.235).

The status of the physical printer appropriate to the currently selected logical printer will be given by *PSLIST after the printer server station number. The status codes currently supported are:

Ready -- this printer is ready to start printing, or that print spooling is available for this printer.

Busy with nnn -- this non-spooling printer is busy printing output from station nnn.

Jammed -- this printer has jammed with paper, has run out of ribbon or some similar event. For a print-spooling printer, the directory %PRINTQ may be full or not found. Jammed printers will not accept any data.

If you are not allowed to use the logical printer which is currently selected for you on a printer server, *PSLIST will not list that station. Note that it is not possible to select a logical printer you cannot use with the commands *PS and *PRINTER; so there are only three reasons why a printer server should not be listed:

1. You may not be allowed to use any logical printers on that printer server. The logical printers may not exist, or they may require users to be logged on the File Server or to have access to a particular account.

2. You may not have changed your logical printer on that printer server since you logged on, and the default printer on that File Server may not be available to you.

3. You may have selected a logical printer on that printer server with *PS or *PRINTER when you had access to it; but the system manager has edited the printer information so that you are no longer allowed to use that logical printer.

Examples:

```
*PSLIST
000.235 Ready
     EPSON       Mac       x NOBANN      LASER       CONDEN      PSCRPT
NONSPL    MCMULT000.251 Ready
    parall    serial    hold      auto
200 Busy with 023
```

Likely Errors:

There are no errors specific to this command.

Associated keywords:

`*PRINTER, *PS`

Compatibility Notes:

Supported by Acorn systems, except that Acorn systems do not support logical printers.

RISC OS Notes:

Not available on RISC OS.

Syntax:*PSLIST

Description:

*PSLIST displays a list of all of the eligible printer servers on the network, i.e. those you are currently allowed to use. For SJ Research printer servers, the logical printers available on each File Server will be listed after the station number.

Logical printers will not be listed if they have been set by the system manager to be non-existent. However, all other logical printers on an eligible printer server will be listed, even if you are not allowed to use them.

If the Econet installation comprises multiple networks, *PSLIST will also display printer servers on other networks, preceded by their network number (e.g. printer server 235 on network 2 will be displayed as **002.235**).

The status of the physical printer appropriate to the currently selected logical printer will be given by *PSLIST after the printer server station number. The status codes currently supported are:

Ready -- this printer is ready to start printing, or that print spooling is available for this printer.

Busy with nnn -- this non-spooling printer is busy printing output from station nnn.

Jammed -- this printer has jammed with paper, has run out of ribbon or some similar event. For a print-spooling printer, the directory %PRINTQ may be full or not found. Jammed printers will not accept any data.

If you are not allowed to use the logical printer which is currently selected for you on a printer server, *PSLIST will not list that station. Note that it is not possible to select a logical printer you cannot use with the commands *PS and *PRINTER; so there are only three reasons why a printer server should not be listed:

1. You may not be allowed to use any logical printers on that printer server. The logical printers may not exist, or they may require users to be logged on the File Server or to have access to a particular account.

2. You may not have changed your logical printer on that printer server since you logged on, and the default printer on that File Server may not be available to you.

3. You may have selected a logical printer on that printer server with *PS or *PRINTER when you had access to it; but the system manager has edited the printer information so that you are no longer allowed to use that logical printer.

Examples:

```
*PSLIST

235 Ready
simple
fancy
nobann
200 Busy with 023
```

Likely Errors:

There are no errors specific to this command.

Associated keywords:

*PRINTER, *PS

Compatibility Notes:

Supported by Acorn systems, except that Acorn systems do not support logical printers.

*PTIME * PDATE * *PDATE2

Syntax:*PTIME | *PDATE | *PDATE2

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:lh

```
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.

Associated keywords:

*GTIME, SETTIME

Compatibility Notes:

Supported on Acorn systems. Level 2 File Servers have only the date; level 3 File Servers have both date and time, and they use the interval timer function in the BBC Microcomputer thereafter -- there exists the possibility that this timer could lose time under certain circumstances.

Syntax: *PUSER

Description

This program prints the name of the user logged on to the station on the screen. This is not followed by a carriage return so more text can be added directly after it. *PUSER can be used in cases of amnesia or to personalise printed output from programs.

User names have a maximum length of 10 characters.

Examples:

```
10PRINT "Hello ";
20  *PUSER
30  PRINT ", you are wonderful !"
```

This is similar to the command ***GUSER** which allows the user name to be read into a string.

Note that, whilst lines 10 and 20 may be combined with a colon, the following text must be on its own line if *PUSER 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("PUSER"), to avoid this problem).

Likely Errors:

There are no errors specific to this command.

Associated Keywords:

*CV, *FS, *GUSER, *I AM

Compatibility Notes

Supported by Acorn systems, but note that some Acorn File Servers support user names of more than 10 characters.

* PUTGET AND * PUTGET2

Utility program
Must be run before a loading user program.

Syntax: *PUTGET |
*PUTGET2

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 OSGBPB filing system call. This results in a speed increase by a factor 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<Return> 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.

Associated Keywords:

BGET, BPUT

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.

* REMOTE

Transient program

Syntax: *REMOTE <station number> | *REMOTE <User Id>

Action with Wild Cards in the User Identifier:

Occurs on the first match, from the top of the list as produced by *USERS (see this Section) downwards.

Description

This utility allows the user to take over one other specified station, or the station where a specified user is logged-on. This station will then echo the screen output from your station. This is useful for demonstrations, but it is a good idea to check that this will not disrupt the other user in the middle of an operation.

Control of a remote station is turned off by the command ***ROFF**, which is not kept in the utilities library but is provided in the network ROM inside the machine. Attempts to take over a second station at the same time will have no effect and it is not possible to take over your own station.

The commands ***PROT** and ***PROTEX** can be used to protect your machine against *REMOTE. In that case, or if a station number that is not logged-on is specified, the *REMOTE machine will wait until <Escape> is pressed.

If the specified user is logged-on at several stations, only one of them will be taken over: 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 (see this Section).

Likely Errors:

Not logged on **Error 174 (AE)**
If a user identifier was specified, and that user was not logged on at the currently selected File Server.

Associated Keywords:

*NOTIFY, *PROT, *ROFF

Compatibility Notes

Supported by Acorn systems.

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:

Circular RENAME **Error 7 (07)**
Caused by an attempt to rename a directory as a sub-directory of itself.

Renaming across discs **Error 176 (B0)**
It is not possible to transfer an item from one disc to another using *RENAME.

Insufficient access **Error 189 (BD)**
The user does not own either the file being renamed, or the directory into which he is trying to rename it.

Bad wildcard **Error 204 (CC)**
The number and types of wild card characters must match in the *RENAME command.

Already exists **Error 196 (C4)**
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.

Syntax: *REROUTE <print job name> <logical printer name>

Action with Wild Cards in File Name:

Occurs on every match.

Description

This command changes the logical printer selected for a print job in the print queue on the currently selected File Server. The user must have owner access to the print job and be allowed to use the new selection for the logical printer. There is no need to give the full pathname of the print job; this command will automatically look in %PRINTQ since it can only apply to files there.

*REROUTE is particular useful in combination with the special logical printer HOLD; as output can be routed to HOLD, inspected to see that it is as required, and then sent to another logical printer with *REROUTE.

Examples:

```
*REROUTE AA03 fancy
```

will change the logical printer selected for the print job AA03 to the printer fancy.

Likely Errors:

FS Unusual Error 46
An attempt to reroute a file or directory, not a print job.

Not authorised to use printer Error 10 (0A)
If the system manager has restricted the use of this logical printer to users with access to a certain account.

Insufficient access Error 189 (BD)
Only an owner can change the logical printer for a print job.

xxxx not found Error 214 (D6)
If either the print job or the logical printer specified does not exist.

Associated Keywords:

*PS, *PRINTOUT, *PRINTER

Compatibility Notes

Not supported on Acorn systems.

Syntax: *RUN <file specifier> | */<file specifier> | *<file specifier>

Action with Wild Cards in File Name:

Occurs on first match (alphabetically).

Description

This command 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:

xxxx is not a file **Error 181 (B5)**
An attempt has been made to *RUN a directory.

Insufficient access **Error 189 (BD)**
The user must have access **R** to the file.

xxxx not found **Error 214 (D6)**
The file does not exist.

Compatibility Notes

Supported on Acorn systems.

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$="MyProg"
```

followed by:

```
SAVE A$
```

will save the current program as **MyProg**.

Likely Errors:

Too short **Error 6 (06)**
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.

xxxx is a directory **Error 181 (B5)**
Caused by attempting to create a file with the same name as that of a directory.

Insufficient access **Error 189 (BD)**
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.

Already opened by xxxx **Error 194 (C2)**
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.

Bad name **Error 204 (CC)**
If the file name contains illegal characters (e.g. **$ % . ^:** except in contexts where they are permitted)

Bad wildcard **Error 204 (CC)**
Use of * or # in the file name.

Entry locked **Error 195 (C3)**
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.

Associated Keywords:

LOAD, *SAVE

Compatibility Notes

Supported on Acorn systems.

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:

Too short **Error 6 (06)**
Caused by an attempt to save a file of less than 16 bytes length. This error is caused if the system manager has set a special option for a user, and its occurrence can be changed using *ENABLE SAVES or *DISABLE SAVES.

xxxx is a directory **Error 181 (B5)**
Caused by attempting to create a file with the same name as that of a directory.

Insufficient access **Error 189 (BD)**
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.

Already opened by xxxx **Error 194 (C2)**
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.

Bad name **Error 204 (CC)**
If the file name contains illegal characters (e.g. $ % . ^: except in contexts where they are permitted)

Bad wildcard **Error 204 (CC)**
Use of * or # in the file name.

Entry locked **Error 195 (C3)**
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.

Associated Keywords:

*LOAD, SAVE

Compatibility Notes

Supported on Acorn systems.

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:

Bad name **Error 204 (CC)**
If the disc name contains illegal characters.

xxxx not found **Error 214 (D6)**
If the disc of that name is not present in the system.

Associated Keywords:

*DIR

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.

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 (see this Section) can be used to turn the edited text back into a BASIC program:

```
*SPOOL TEXTFILE
LIST
  .
  .
  .
*SPOOL
```

Likely Errors:

xxxx is not a file **Error 181 (B5)**
Caused by attempting to create a file with the same name as that of a directory.

Insufficient access **Error 189 (BD)**
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.

Too many files open **Error 192 (C0)**
There is a limit to the number of channels available, normally 5 or 6.

Already opened by xxxx **Error 194 (C2)**
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.

Bad name **Error 204 (CC)**
If the file name contains illegal characters (e.g. **$ % . ^:** except in contexts where they are permitted)

Entry locked **Error 195 (C3)**
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.

Associated Keywords:

*EXEC, *PRINTOUT

Compatibility Notes

Supported on Acorn systems

Syntax: *STATEMENT

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 (see this Section). 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 in Section 5.4.

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.

Associated Keywords:

*ACCOUNT, *CREDIT, *DEBIT, EDITPASS

Compatibility Notes

Not supported on Acorn systems, since these do not run an accounting system.

Syntax: *STATIONS [<network number>]

Action with Wild Cards in Network Number:

Wild cards prohibited.

Description

This utility displays a list of all machines coonected to the Econet network, that are currently switched on. Each line is of the form:

<station number> <machine type> <version number>

The machine types include:

- Acorn BBC micro
- Acorn Master
- Acorn E.T.
- Acorn System 5
- Acorn System 3/4
- SJ Research File Server
- SJ Research Z80 CP/M
- SJ Research IBM
- SJ Research SCSI

*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
  135     SJ Research SCSI
   87     Acorn BBC micro
   32     SJ Research Z80 CP/M
    8     Acorn Master
    2     Acorn 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.

Associated Keywords:

*CV, *FSLIST, *PSLIST

Compatibility Notes

Supported by Acorn systems.

Syntax: *TIME

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.

Associated Keywords:

*GTIME, *PTIME, *PDATE, SETTIME

Compatibility Notes

Supported on Acorn systems. Level 2 File Servers have only the date; level 3 File Servers have both date and time, and they use the interval timer function in the BBC Microcomputer thereafter -- there exists the possibility that this timer could lose time under certain circumstances.

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 OSGBPB 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 Sections 5.5 and 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.

Associated Keywords:

*DUMP

Compatibility Notes

Supported by Acorn systems.

Syntax: *UNPROT

Description

This program removes the protection that has been set up by ***PROT** from the station at which it is entered. *PROT prevents other stations sending operations which affect your station. Protection will also be removed by a <Ctrl-Break>.

If some operations from other stations are required but not others, an extended version of this command is given by ***PROTEX**.

Likely Errors:

There are no errors specific to this command.

Associated Keywords:

*PROT, *PROTEX

Compatibility Notes

Supported by Acorn systems.

Syntax: *USERS

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 specifed user performed a filing operation.

Note that if a user was logged-on, and switches off his machine without typing *BYE, he will still appear in *USERS until someone else turns that machine back on and logs-on from it.

The users listed may include *-SYSTEM-* and *-SPOOL-* at station 0. These are used by the system to carry out print spooling and other system operations. These users do 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.

Associated Keywords:

*CV, *GUSERS, *PUSERS, *STATIONS

Compatibility Notes

Supported on Acorn systems, but the user list is not sorted in the same way, and is essentially random in order.

Syntax: *VERS

Description

This program displays the current version number of the File Server. This will be of the form:

SJ 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.

Associated Keywords:

*FSLIST

Compatibility Notes

Supported by Acorn systems, but the displayed number will be different.

Syntax: *VIEW <station number> | <user identifier>

Action with Wild Cards in the User Identifier:

Occurs on first match (chronologically).

Description

This utility allows a station to make a complete copy of a remote station's screen. The prompt will then be returned.

Examples:

```
*VIEW FRED
```

Will copy the screen of the station number that FRED is logged on to. If there is more than one occurence of FRED in the user list, then the station which he most recently used to access the File Server will be copied.

```
*VIEW 1.235
```

will copy the screen of station number 235 on network number 1.

If the remote machine is in a screen mode which takes more memory than the screen mode in the local machine then the error 'Mode x' will be returned, where x is the screen mode of the remote machine. After the 'Mode x' error the value of x can be read by OSWORD A=&13 with function code 10. For example:

```
10 DIM S% 15
20 S%?0=10
30 X%=S%:Y%=S% DIV 256
40 A%=&13:CALL &FFF1
50 PRINT"Screen mode is ";S%?1
```

Likely Errors:

Inactive **Error &A2**
If the remote station is not on the network.

Mode? **Error &A4 Mode**
If the remote station is a fileserver or some other machine with a different screen layout.

PROTed **Error &A2**
If the remote station has issued a *PROT command.

Not logged on **Error &AE**
If the user identifier specified is not currently logged on.

Mode x **Error &AD**
If the screen mode of the remote station uses more memory than the screen mode you are currently in.

Associated Keywords:

*PROT, *REMOTE

Compatibility Notes

Supported by Acorn systems.