

About This Remade Document

Title: Advanced NFS ROM (BBC)
Last remade: 31-Mar-2025
Repository: <https://github.com/acheton1984/AcornDocsRemade>

This is not quite the beautifully "remastered" document as seen elsewhere, but it is intended to be very similar to the original in an easy-to-read and searchable PDF format.

Reconstruction Notes

Source files: <https://www.stardot.org.uk/forums/viewtopic.php?p=421794#p421794l>

- Recreated from the original EDIT document by capturing the printer output from MAME.
- Spelling errors have been corrected where spotted.
- A title and remade document history page have been added, along with a link to this repository.

Disclaimer

The underlying content is reproduced or adapted from the original documents for educational purposes and ease of access.

All rights remain with their respective owners.

Remade from an original EDIT-format document. This PDF is hosted at:
<https://github.com/acheton1984/AcornDocsRemade>

This document describes ANFS for the BBC microcomputer.

Last changed 22/05/85 by Brian Cockburn.

1 New features implemented.

1.1 Version.

This is now version "NFS 4.08.61".

1.2 Help.

The *HELP command now shows the current version number and also shows the station number on a line which is identical to that displayed at BREAK time. There are now two sub categories, "Net" for those commands which are part of the network filing system, and "Utils" for filing system independent utilities and commands.

1.3 Local buffering.

The new ROM will buffer data from all open files in RAM in the I/O processor. This will mean that all uses of the "osbget" and "osbput" calls will be significantly faster since only a small proportion (2%) will actually need to communicate over the network with the file server. The buffering code will use 256 byte buffers on a "dynamic" basis. This means that there may be more than one buffer allocated per file channel. The number of active channels capability will be increased from the current eight to sixteen. This will mean that higher performance file servers can be implemented without need to change the ANFS. The limit of sixteen will also permit the user to maintain valid context and open files on more than one file server at once. The current limit of eight channels is built in to both the file server and the original NFS.

For this "dynamic buffering" to take place a number of buffers are allocated dynamically to only those files that are open. The number of buffers is actually variable, but if there were five then a typical example would be if a single file was opened, then as it is read the first five pages of the file are sequentially put in to the buffers so if they are referred to they are available instantly. When the sixth page is requested then the page that was referred to least recently is "thrown out", this is the way that it will tend to function as cache. This means that files are to a small extent cached by being in the buffer(s). If two files were open then the buffers would be shared on a use basis so one file was being use twice as much as the other then it would have about twice as many buffers.

The number of buffers will be set to five initially. By having ROMS other than the ANFS ROM plugged in then more than five buffers may be available. The following ROMs will result in more buffers being available; Acorn DFS, Acorn TFS, and Acorn ADFS, with six, nine, and eleven, respectively. The maximum number of buffers is always sixteen.

The algorithm for determining which buffer should be marked "Least Recently Used" (LRU) is as follows; if the buffer has nothing in it then it will be come LRU, if there are no buffers in this state then the buffer adjacent to the buffer being accessed will be marked, this will not occur if the buffer has data in it but is not full. This situation of a partially full buffer will prevent the "leading edge" of a file being "thrown out". If multiple files are open then some algorithm will determine how much space is used for each file ie, if one file is "getting a lot of action" then it may have "more than it's fair share" of buffers allocated to it.

The translation of user handles to file server handles is now more complex than before. So the OSWORD call to set the context handles works in a slightly different way. If the user handle is not open or is not a directory then no change will occur to the handle being written. When this happens rather than issuing a channel error the reason code in the OSWORD block will be changed to zero to indicate failure. Also an OSARGS call will be available to return the file server handle (and number) for a given user handle.

1.4 Extra ANFS star commands incorporated in the ROM.

1.4.1

Cdir <dir> (<number>)

Where the size is quoted in entries and a directory big enough to hold that many entries will be created. The size range available is from one to 245, any number outside this range will give an error. If the optional parameter is not given then the default of 19 will be used, this creates a directory of length &200.

1.4.2

Flip

The FLIP command will simply exchange the CSD and CSL. This is a way of selecting the library as your CSD, this is useful when files which must be LOAded (via the OSFILE mechanism) are to be made public, and software must be able to access them easily. It is unwise to use the *DIR or *LIB commands in the "Flipped" state, the user should FLIP back first.

1.4.3

FS (<stn. id.>)

The FS command will change the file server number. This allows a user to be logged on to two or more file servers at one time and to change between them. Any open files will be ensured to the current file server before the number is changed. If no argument is given ie. just *FS<CR> is typed then the current file server number will be reported.

1.4.4

I am (<stn.id.>) <user id.> ((:<CR>)<password>)

This command is essentially unchanged from previous versions of the NFS but now accepts <DELETE> and control U during the "invisible" part and will delete either the last character typed or the entire "invisible" part respectively. It is possible for this command to display the warning message "Data Lost" if data which had been written to a file was still buffered and was not able to be written to the file server. This could be caused by the file server having been restarted.

1.4.5

Lcat (<dir>)

Catalog the current library. Can also take an optional argument for a path from the currently selected library eg. *LCAT fonts<CR> will catalog a directory called "fonts" in the currently selected library (CSL). This facility is useful since, for example, the command *FONTS.ITALIC is looked up in the currently selected library.

1.4.6

Lex (<dir>)

Examine the current library. Also takes an optional argument, see *LCAT.

1.4.7

Pass (:<CR>)<old password> <new password>

Will now accept a ":" in the middle of the command like *I AM now does so that passwords may be hidden. It should be noted that although the ":" may appear anywhere in the line it would be most useful to have it before both the old and new passwords. Whilst the "invisible" part is being typed control U deletes the entire "invisible" part and <DELETE> deletes the last character.

1.4.8

Remove <object>

This command is identical to *DELETE with the exception that it does not give the error "Not found" if there is nothing to delete. This makes it most useful in a program since it will not cause an unexpected error.

1.4.9

Wipe (<dir>)

Will offer each unlocked file or directory in the directory for deletion with a (Y/N/?) if "?" is typed the full object information will be printed followed by (Y/N). If "Y" is typed then the file or directory will be deleted. If anything else is typed the object will not be deleted.

1.5 Extra Utils star commands incorporated in the ROM.

These commands may be issued when the ANFS is not the current filing system.

1.5.1

Close

Will close all open files on the current file server. This is the same as a CLOSE#0. Files (and directories) open on other file servers will not be closed. This command does not close open directories.

1.5.2

Dump <filename> (<offset> (<address>))

This dumps the file in hex and ASCII in a format suitable for the current screen width. The optional offset parameter is the number of bytes (in hex) to skip before starting to dump. The address parameter is the address (in hex) to use as the address of the first byte in the file to be displayed on the left hand side of the dump listing, the default value for this is the files' load address.

1.5.3

Pollps (<stn.id.>|<ps type>)

The POLLPS command will show the currently selected printer server number and the currently selected printer type, eg.

```
>*POLLPS
Printer server is 235 "PRINT"
    235 is ready
>_
```

Following the printer server number will be a list of all the printers on the network and the current status of each. The possible status conditions are either "ready" which means the printer is ready for use or has timed out the current user, "busy" which means the station shown is currently using that printer or "jammed" which means that the printer server has characters in it's buffer but has been unable to send characters out to it's printer for some period of time, this would usually be caused by the printer being off-line or unable to accept characters for some reason. It should be noted that this command does not alter the printer server number, the state of FX5, or the state of FX6.

If the command is followed by a station number then only the status of that station will be shown, no status will be shown if the station is not operating as a printer server. The command can also have a textual argument eg. "DAISY", "LINE", "DRAFT", or "LASER". If this textual argument is supplied then only printers of that type will have their status listed. This allows the users to only examine the status of printers that they are interested in.

1.5.4

Print <filename>

This implements the following approximate BASIC program ;

```
C=OPENIN(file$):REPEATVDUBGET#C:UNTILEOF#C:CLOSE#C
```

The main use of this is to display SP00Led sequences of text and graphics sent to the screen.

1.5.5

PS (<stn.id.>|<ps type>)

The command *PS followed by a number behaves in the exact same way as the loaded transient command PS as supplied with normal NFS (and File servers), ie. it sets the printer server station number to the one supplied. If a textual argument is given then the printer server will be set to the number of the first printer of that type to be found "ready", if no printer is found to be ready then the printer server number will remain unchanged. If no argument at all is supplied then the printer server number will be set to the first "ready" printer. When a textual argument is supplied it will become the currently selected printer type, this type will be used when *PS is issued with no argument. Note the power-up default type is "PRINT" to which all printer servers respond. The command *POLLPS with no argument also uses this currently selected type. If printing is taking place when the PS command is issued then the error message "Printer busy" will be issued and the printer server number will not change. Again it should be noted that this command does not affect the states of FX5 and FX6.

1.5.6

Prot (<prot type>) ...

With no argument supplied this will protect against all operations, if any arguments are given then only those types of operations will be protected against. Note that this can have multiple arguments eg. *PROT PEEK POKE<CR>.

1.5.7

Type <filename>

Is the same as PRINT but treats <CR>, <LF>, <CR><LF>, or <LF><CR> as a line terminator and does a newline function. This command is for displaying text files.

1.5.8

Unprot (<prot type>) ...

With no argument supplied this will unprotect against all operations, if any arguments are given then only those types of operations will be unprotected against. Note that this can have multiple arguments eg. *UNPROT POKE JSR<CR>.

1.6 Printing.

After a *FX5,4 has been issued the issuing of a VDU2 or control B will cause the status of the current printer server to be examined. If this status is "jammed" or "busy" then an error (BRK) will be generated to that effect, then the VDU2 will be cancelled. If any type of error occurs during either transmission or reception from the printer server then, as before, an error will be generated but now the VDU2 will be cancelled. For more information see the section on errors. Issuing a VDU3 whilst not printing will have no effect. Issuing a VDU2 whilst printing will send any data which has been "printed" but buffered locally to the printer server, this will work even if there is no data to send. This is most useful since this act of transmitting to the printer server will reset it's timeout. If the <BREAK> key is pressed whilst printing is taking place then all characters in the printer buffer will be sent and the printing will be terminated (, a VDU3 will be "faked").

1.7 Extra interfaces.

An OSWORD call is available for reading and writing the default printer server type. This type is a six character ASCII string not terminated with a <CR> but space filled to six characters. Other extensions will permit the reading of the handle associated with the last error, if there was one. There is one to read how many characters are in the network printer buffer, and to determine the local network number and to translate external station addresses to local ones.

1.8 Enhancements to the filing system.

1.8.1

Write only files

File servers now support the notion of files which are write only with public access. These files will have access strings such as "WR/W" or "LWR/W" the reason is to support simple "mail" schemes with some privacy. Write only files should be opened for update and will give an error if an attempt is made to read data (either with OSBGET or OSGBPB). It should be noted that there will be no buffering of write only files. Note that this function requires files servers of the following version or above;

"Level 2 Version 1.05" or "Version IV.05".

1.8.2

OSFILE

The new interface is called "create", this behaves exactly the same as "save" except that no data is transferred. This means that files can be created with large lengths without the necessity of transferring large amounts of data. Note that this is useful since to create a file of 10000 bytes in length would have required a save eg. *SAVE FILENAME 0 10000<CR> which would "crash" the BBC machine because in the process of saving the data certain "read sensitive" locations are read. These locations include data transfer registers of some I/O devices eg. Tube, ADLC, and FDC. Note that this function requires files servers of the following version or above;

"Level 2 Version 1.02" or "Version IV.03".

1.8.3

OSARGS

All calls to ensure files now function and ensure open file(s) to the file server. The call which returned the version of NFS returns the same value as NFS 3.40 and NFS 3.60.

A new function, expressed from BASIC as EXT#= is implemented. This can either increase or decrease the length of a file. Note that this function requires files servers of the following version or above;

"Level 2 Version 1.02" or "Version IV.03".

The amount of disc space allocated to a file can now be read, this is greater than or equal to the current extent. An interface has also been provided to provide for Z80 based software (CP/M,) the file server handle for a particular file, users are advised NOT to use this handle since its use could result in lost data.

1.9 Error messages.

Both the "Not listening" and "No reply" error messages will have the station number added to them so they will now become "Station nnn not listening" and "No reply from station nnn". The "Channel" error will now be followed by the channel that was found to be in error eg. "Channel 99". If any sort of error occurs during a "random access" operation (OSBGET, OSBPUT, OSARGS, OSFILE, and OSGBPBP) then information regarding the channel that had the error will be appended to the error message for example if the file server was very busy a message like : "Station 254 not listening on channel 32" might be caused. It should be noted that the channel associated with the error may not have been the channel the operation was attempted on. If for instance an OSBPUT is attempted then a buffer may be required, if this means that the previous contents of the buffer need to be written back to the file server then the error may occur on the channel associated with that buffer.

The fatal error caused by the OSWORD call now produces the message "Remoted" and may have channel information added if the error occurred during a "random access" operation.

There are now many new errors, "No!" which occurs when an attempt is made to *RUN a file with a load address of &FFFFFFxx and an execute address which is not &FFFFFFFF. The expanded printer interface now has the extra errors "Printer busy" , "Printer jammed" , and "Station not present". There is an error "Syntax" for commands which are recognized but have the wrong syntax (missing parameters). The ANFS now maintains a checksum on its private workspace and this is checked before any random access operation, the new error is "Bad net sum". If a file is opened on a particular fileserver then it can only be accessed when using that file server. If an attempt is made to use the file whilst logged on to another file server then the error "Channel nnn not on this file server" will be issued. The "Bad hex" and "Bad number" errors occur if a number was expected and non-hexadecimal or numeric characters were encountered. The errors "Bad station number" and "Bad net number" are issued from, for instance, the *I AM command. "Bad parameter" will be caused if a numeric argument is out of range. "Write only" is the error from attempting to read a write only file. "No more FCBs" will be given if an attempt is made to open more than 16 objects at one time.

1.10 A URD reference point.

Any object reference that starts with "&" is assumed to refer to a pathway from the URD not the CSD. To a certain extent this is compatible with the ADFS since ADFS interprets "&" to mean the same as "\$". This will make access to files in non-local parts of the users' directory structure easier.

1.11 Compatibility with DFS based software.

If an object reference starts with ":0." ie. an explicit disc reference then this will be translated to "&.0.". This means that disc software which references specific files can be made to work under the ANFS by creating a directories "0" and "1" in the users root directory. This does prevent users using single character titles for discs, but this was an unlikely event in any case.

1.12 Additional library functionality.

To be compatible with the ADFS, ANFS will adopt the convention that any file which is *RUN <filename>, */<filename>, or *<filename> will be treated as *EXEC <filename> rather than *RUN <filename> if it has an exec address of &FFFFFFF. This means that EXEC files for commonly used sequences can be stored in the library. There is also a "User Library" which is searched after the CSD and before the CSL. This new library can therefore be used to over-ride normal library functions as well as to extend the users personal library. The main advantage of this is that users who need nonstandard libraries no longer need to duplicate sections of the main library. This "User library" is "&.Library", so the user should create a directory called "Library" in their root directory if they wish to take advantage of this facility.

1.13 Time and Date.

The time and the date as known by the file server have always been readable by the user, via the OSWORD call. There now exists a second OSWORD call which reads the same information in a different (standard) format. This format returns the information in BCD rather than packed binary.

1.14 IO processor address space.

The IO processor normally accessed using addresses between &FFFF0000 and &FFFFFFF. For compatibility with future products this space has been increased to 128K bytes. So now the IO processor memory can be addressed between &FFFE0000 and &FFFEFFFF as well. It is recommended that this secondary area be used to refer to the screen RAM only.

1.15 Automatic Bootstrapping.

During a logon the normal course is to establish the users startup option and act on it. Now any action which may have taken place will be suppressed if the <CONTROL> key is held depressed during the boot sequence. Note that this suppression does not occur if the logon is a result of a "<SHIFT><BREAK>".

1.16 Re-tries.

The defaults for the number of retries for several operations are now "user adjustable". The number of transmit retries is adjustable from one to 255 if zero is set then transmit will try "for ever". Since some operations are normally in-ESCapable after 255 tries the operation does become ESCapable. The default for transmit is 255. The operation *I AM <user id.> and VDU2 have a "machine peek" prior to determine the existence of the destination station, for this type of transmit the default is ten. When waiting for a reception the receive block is checked some number of times and if the reception has not occurred then the "No reply" error is issued. The default number of times to check is 40. These values are accessed via an OSWORD call.

1.17 File server / Bridge net number translation.

To support the use of bridges there is a call to translate a net number given by a remote station to one relative to the current station. This will be via OSWORD. This will mean that *VIEW, *REMOTE, and *NOTIFY will work for all cases in multi net configurations. The same OSWORD will also return the local net number and the version number of the bridge code. In the normal way if this call should fail for some reason the error will be indicated in the data returned to the user.

The commands "*I AM", "*FS", and "*PS" have been changed so that if a full station number, including network number, is given then the network number is compared to the local net number. If the net number given is the same as the local net number then the network number will be treated as zero. This allows the use of global numbering without worry.

1.18 Peaceful co-existence and the like.

Since there is more than one version of the ANFS a method has been devised to prevent the incorrect version from running in any machine. This is done by checking the version of the operating system. If the wrong operating system is detected the ROM will pretend it is unplugged and then the machine starts up in the normal way and will completely ignore the bad ROM.

To prevent problems which would otherwise occur when the ANFS ROM and the DNFS ROM are both plugged in to the same machine an automatic mechanism will "frugal" the NFS part of the DNFS ROM so that the machine will function correctly. This is necessary if a machine is to have both Econet (ANFS) and DFS.

1.19 Entry of numbers in hexadecimal.

At any place a decimal number is expected a hexadecimal number may be entered if it is preceded by the "&" character, this is the same format as BASIC uses to enter hexadecimal. Note that where an entry in hexadecimal is expected the "&" character should not be used.

1.20 Tube host code.

The ANFS ROM has the Tube host code the same as the DNFS had. The operation of the Tube code does not rely on the existence of the Econet hardware.

1.21 Events on reception.

There is now a new event with number 254 which is the Econet receive event. This event is enabled with *FX52,150 and disabled with *FX52,100. When enabled an event will be generated by the completion of a successful reception. The Y register will have the receive control block number that was used by the reception.

2 Technical details of the new features.

2.1 *OSWORD A=&E, Read the time.*

X and Y point to a control block the first byte of which is the reason code. If the reason code is four then the time and date are placed in the control block in the following format;

XY + 0 => Year since 1900, in BCD
XY + 1 => Month, in BCD
XY + 2 => Day of the month, in BCD
XY + 3 => Day of the week, always zero
XY + 4 => Hours, in BCD
XY + 5 => Minutes, in BCD
XY + 6 => Seconds, in BCD

2.2 *Changes to OSWORD A=&10.*

Transmit has been changed slightly to permit the transmission of no data, ie. a transmission of zero length. This is inter-operable with previous versions of the NFS. Due to restrictions of the Tube zero length transmissions are only available from the I/O processor, since the source of zero length transmissions is unimportant this is not a major problem. If a zero length transmission is attempted from the Tube side then a 64K transmit will occur. This is the same as previous versions of the NFS.

2.3 *Changes to OSWORD A=&13.*

There is now a defined "fault" return where for some reason the OSWORD has failed, this is indicated by the reason code being returned as zero. Calls with reason codes 0,1,6, and 7 can fail if the ANFS is not the current filing system. Call 17 can fail if contact with a bridge fails whilst trying to determine the local network number. Call 7 could fail if any one of the three handles were not for open directories on the current file server.

The new calls are as follows, 11 is "Read error channel number" returns the channel of the last error in byte 1 of the OSWORD block. Note byte 0 is the reason code. An error channel of 0 means no channel was associated with the last error.

Call 12 and 13 read and set the printer server type text referred to in in section 1.5.5. The printer server type is read from or set to the bytes 1 to 6 of the OSWORD block. The printer server type is always stored as upper case text so reading and setting will not always return the same information.

Call 14 returns the number of characters in the local printer buffer in byte 1 of the OSWORD block. Call 15 and 16 read and set the delay and retry count values. The normal transmit retry count is in byte 1, the receive poll count is in byte 2, and the machine peek retry count is in byte 3. The last new call is 17, it returns the local network number in byte 1, byte 2 is not touched, and byte 3 is translated as follows;

```
DEF FNtranslate(number)
IF number = 0 THEN = file_server's_network_number
IF number = local_network_number THEN = 0
= number
```

2.4 Changes to OSFILE.

The call with A=7 now does the operation "Create", which is the same as "Save" but no data is transferred. Entry and exit conditions are the same as "Save". Note that this function requires file servers of the following version or above;

"Level 2 Version 1.02" or "Version IV.03".

2.5 Changes to OSARGS.

If A=3 then the function expressed in BASIC as "EXT#channel=value" is performed. The handle is in Y and the value is in a four byte block of memory pointed to by the X register. Note that this function requires file servers of the following version or above;

"Level 2 Version 1.02" or "Version IV.03".

If A=4 then the amount of space currently allocated to the file is returned at X the same as for A=0 or A=2.

If A=&80 then the following information is returned in the four bytes in zero page pointed to by X. The first byte has the external or file server handle for the object whose user handle is given in Y, the file server number is given in the second and third bytes and an ANFS internal status byte is returned in the forth byte. The format for this status byte is as follows;

```
Bit 0 is the sequence number
Bit 1 is the "known to be a dir" flag bit
Bit 2 is the "thought to be URD" bit
Bit 3 is the "thought to be CSD" bit
Bit 4 is the "thought to be CSL" bit
Bit 5 is the "current context" bit if the object is a directory
Bit 5 is the "write only" bit if the object is a file
Bit 6 is the EOF error bit, set if next BGET will fail
Bit 7 is the write flag, 1 means writable to
```


The call with A=&FF will ensure the file whose handle is in Y to the file server, note that this does not close the file but merely ensures that the data associated with the file is up to date in the file server. If Y=0 then all files will be ensured. This is the same as DFS and ADFS.

2.6 Errors.

There are several new errors, they are listed below in numerical order together with their descriptions.

2.6.1

0, "Remoted".

This error, which should be regarded as "fatal" is usually a result of another station running the REMOTE utility. It can also be caused by using an OSWORD call on another station.

2.6.2

&93, 147, "No!".

The "No!" error is issued in response to an attempt to *RUN, */ , or * a file with a load address of &FFFFFFxx and an execute address which is not &FFFFFFFF (see section 1.12). This feature is mainly to permit the filing system to distinguish data files from code or control files. It is also of interest that the five bytes of the exec and load address that are "don't care" could be used to store other information, eg. the time, or the file type.

2.6.3

&94, 148, "Bad parameter".

This error means that the number supplied to *CDIR was not between one and 255.

2.6.4

&A4, 164, "Station <stn. id.> not present".

Two operations, *I AM and control B, now precede their normal actions by a machine peek immediate operation. This machine peek is not protectable against, it is normally used to verify the existence of the station you are attempting to contact. The retry count on this transmission is low so that failure is almost immediate, this is to prevent long delays after mistyping. If a "Not listening" or "Net error" causes the transmission to fail then this "Station nnn not present" error will be reported, if a "No clock" or "Line jammed" causes the error then it will be reported as such. Note that this can fail if the station in question is using its' floppy discs at the time.

2.6.5

&A6, 166, "Printer busy" and &A7, 167, "Printer jammed".

After a control B the printer server is asked what its' current state is. It can return one of three states, Ready, Busy, or Jammed. If the the ready status is returned then the print will start in the normal way. However if busy or jammed is returned then this is reported and the control B is cancelled.

2.6.6

&AA, 170, "Bad net sum".

The ANFS ROM keeps a checksum on the contents of its' third private page, the page used to contain open file information when it is not the current filing system and other information about the current printer type. This checksum is verified on most operations and if it is found to be in error then this error will be generated. The checksum can only be reset by a control <BREAK>.

2.6.7

&B0, 176, "Bad rename".

The error "Bad rename" is caused by attempting to rename using one (but not two) pathnames starting with "&", the URD. Attempting to use "&" as one or other pathname. Only supplying one name. Note that it is legal to use "&" in both pathnames.

2.6.8

&B7, 183, "Outside file".

This error occurs if an attempt is made to move the file pointer outside the extent of a file which is only open for input, note that this will also have the channel on the end eg. "Outside file on channel 32". The *DUMP will also give "Outside file" if an attempt is made to start the dump past the end of the file, this will not leave the file open.

2.6.9

&C0, 192, "No more FCBs".

The filing system has 16 File Control Blocks or FCBs. One block is used for every open file or directory. If files are open at more than one file server then it is possible to run out of FCBs with which to open more files, this error will result from attempting to open the 17th object.

2.6.10

&D0, 208, "Bad station number".

Station numbers must be in the range 1 to 254 anything else is illegal.

2.6.11

&D1, 209, "Bad net number".

Network numbers must be in the range 0 to 254 anything else is illegal.

2.6.12

&D4, 212, "Write only".

An attempt was made to read from a write only file. Either OSBGET or OSGBPBB can cause this error.

2.6.13

&DC, 220, "Syntax".

The "Syntax" error is issued if a command is issued with incorrect syntax this is usually because the command requires some sort of argument and one has not been supplied eg. *ACCESS or *PASS. The other case for a "Syntax" error is *REMOVE with two parameters, the reason for this is to prevent remove from being confused with rename.

2.6.14

**&DE, 222, "Net channel", "on channel",
and "not on this file server".**

Simple channel errors occur when a handle which is outside the normal range (&20 to &2F) or is not open is used in a random access operation (OSBGET, OSBPUT, OSARGS, OSGBPB, or OSFIND). If the handle is valid, ie. belongs to an open object, but that object was opened on another fileserver then the text "not on this file server" will be appended to the channel message giving one like "Net channel 32 not on this file server".

If an error of any kind occurs during any random access operation then the channel concerned with the operation will be reported on the end of the error string like this "No reply from station 254 on channel 32" this channel can be read with OSWORD A=&13. Note that the channel shown may not be the channel that the operation was attempted on, this could happen if an OSBGET was attempted and to read the required data associated with another file had to be ensured and the error occurred on that operation. If errors like "No clock" occur during random access operations then the operation is re-startable, this is useful if a large quantity of network traffic prevents file server access for the duration of a timeout period.

2.6.15

&F0, 240, "Bad number".

This means that a character other than "0" - "9", " ", or <CR> was encountered when a number was expected.

2.6.16

&F1, 241, "Bad hex".

This means that a character other than "0" - "9", "a" - "z", "A" - "Z", " " or <CR> was found after the "&" when a number was expected. This could occur in *I AM, *PS, *POLLPS, *FS, or *CDIR.

2.6.17

&FC, 252, "Bad address".

It means the address given for either of the optional arguments contained a non hexadecimal character or the address was greater than &FFFFFFFF.

2.6.18

&FD, 253, "Bad string".

A bad string is one with an odd number of quotes. Some commands need to process the given arguments and will respond with this error, eg. *LIB "fred".

3 Description of syntax items.

3.1

<address>

Is a 32 bit address in hexadecimal. If more than eight significant digits are given then the error "Bad address" &FC will be given. If a non-hex character is encountered then the "Bad hex" &F1 error will occur.

3.2

<CR>

This is used to represent the RETURN key or character &0D, decimal 13. This character is usually called "Carriage Return".

3.3

<dir>

A directory specification which may include a path name, and as in <filename> may be abbreviated.

3.4

<filename>

A file name. This name may include directory information ie. it may be a path name like "basicprogs.myprog". The file server will also accept abbreviations for filenames and directory names. By abbreviating a name with an "*" the number of keystrokes used is reduced. For example if the above was given as "ba*.m*" then the first file starting with "m" in the first directory starting with "ba" would be used. Note that this use of "*" does not imply multiple files merely the first to be found.

3.5

<LF>

This is used to represent the character &0A, decimal 10. This character is usually called "Line Feed".

3.6

<number>

Is a decimal number, this number will be converted to a single byte and therefore should not be greater than 255.

3.7

<offset>

Is a 32 bit address in hexadecimal. If a non-hex character is given then the error "Bad hex" &F1 will be given. If the value is greater than the length of the file then the error "Outside file" &B7 will be caused.

3.8

<password>

A password for use with the file server, this is a six character string which must start with a non-numeric.

3.9

<prot type>

A protection type, used to set or clear the protection mask byte. May have one of the following values; "Halt", "Peek", "Poke", "Jsr", "Proc", or "Utils".

3.10

<ps type>

Printer server type, may be any string upto six characters in length, examples; "BNY", "QUME", "DAISY", "COLOUR", "LINE". Note that if more than six characters are given then only the first six will be used.

3.11

<stn. id.>

Is a full station number and may include a network number. The network number precedes the station number and is delimited by a ".". If the string ".nnn" is given then the network number will be set to 0 and the station number to nnn. If the string "nnn." is given then the network number will be set to nnn and the station number will be left unchanged. Strings like "nnn.mmm" will set both numbers. The special case "." will zero the net number leaving the station number unchanged.

3.12

<stn number>

Is the local station number. It must be in the range 1 to 254.

	Table of contents	Page
Section 1	New features implemented	2
1.1	Version	2
1.2	Help	2
1.3	Local buffering	2
1.4	Extra ANFS star commands incorporated in the ROM	3
1.5	Extra Utils star commands incorporated in the ROM	5
1.6	Printing	8
1.7	Extra interfaces	8
1.8	Enhancements to the filing system	8
1.9	Error messages	10
1.10	A URD reference point	10
1.11	Compatibility with DFS based software	11
1.12	Additional library functionality	11
1.13	Time and Date	11
1.14	IO processor address space point	11
1.15	Automatic Bootstrapping	12
1.16	Re-tries	12
1.17	File server / Bridge net number translation	12
1.18	Peaceful co-existence and the like	12
1.19	Entry of numbers in hexadecimal	13
1.20	Tube host code	13
1.21	Events on reception	13
Section 2	Technical details of the new features	14
2.1	OSWORD A=&E, Read the time	14
2.2	Changes to OSWORD A=&10	14
2.3	Changes to OSWORD A=&13	14
2.4	Changes to OSFILE	15
2.5	Changes to OSARGS	15
2.6	Errors	16
Section 3	Description of syntax items	21