File Commands

Functions and instructions to handle file access

Load Image PATH$, INDEX, TAGS$
Load an image into a screen
Parameters:
PATH$: Path to the image to load
INDEX: Index of the screen to create. If ommited the image will be loaded in the current screen
TAGS$: List of tags indicating how to load the image. "#left", "#center", "#right" aligns the image horizontally, "#top", "#middle", "#bottom" aligns the image vertically, "fit" resizes the image to fit the screen, "paste" does not resize the image

Load Iff PATH$, INDEX, TAGS$
Load an IFF image into a screen (deprecated, use "Load Image")
Parameters:
PATH$: Path to the image to load
INDEX: Index of the screen to create. If ommited the image will be loaded in the current screen
TAGS$: List of tags indicating how to load the image. "#left", "#center", "#right" aligns the image horizontally, "#top", "#middle", "#bottom" aligns the image vertically, "fit" resizes the image to fit the screen, "paste" does not resize the image

Save Iff PATH$, INDEX
TODO! Save an IFF image out of a screen (deprecated, use "Save Image")
Parameters:
PATH$: Path to the image to save
INDEX: Index of the screen to save. If ommited the image will be saved from the current screen

Load PATH$, INDEX
Load a previously saved memory bank, or a set of banks
Parameters:
PATH$: Path to the bank(s) to load
INDEX: Index of the bank to load into. If not specified, the bank will be loaded at the same number it was saved

BLoad PATH$, INDEX
Load a binary file into a bank
Parameters:
PATH$: Path to the binary file to load
INDEX: Index of the bank to load into. This bank must be of "Data" or "Work" type.

BSave PATH$, START, END
Save the content of a bank to a binary file
Parameters:
PATH$: Path to the binary file to create
START: Memory address of the data to save. Use the Start() function to obtain the address
END: End of the memory zone to save. Use the Start() function to obtain the address and add the desired length

Save PATH$, INDEX
Save a memory banks to a file. This file can later be loaded with "Load"

Parameters:
PATH$: Path to the file to create
INDEX: Index of the first bank to save

Save To PATH$, START, END
Save one or several memory banks to a file. This file can later be loaded with "Load"
Parameters:
PATH$: Path to the file to create
START: Index of the first bank to save
END: Index of the last bank to save

DFree
Returns the amount of free space of the disc pointed to by the current path
Value returned:
integer: The amount of free space

MkDir PATH$
Creates a new directory
Parameters:
PATH$: Path to the new directory to create

Open Random CHANNEL, PATH$
Open a random access file
Parameters:
CHANNEL: Number of the channel
PATH$: Path to the file to open

Open Random CHANNEL, PATH$
Open a file for input only
Parameters:
CHANNEL: Number of the input channel
PATH$: Path to the file to open

Open Out CHANNEL, PATH$
Open a file for input only. The file is replaced by the new one
Parameters:
CHANNEL: Number of the input channel
PATH$: Path to the file to open

Append CHANNEL, PATH$
Open a file for output, and add content at the end of it
Parameters:
CHANNEL: Number of the channel
PATH$: Path to the file to open

Assign NAME$, PATH$
Assign a name to a file or device
Parameters:
NAME$: The name to assign
PATH$: The path or drive name to assign it to

Field CHANNEL, LENGTH, FIELD$
Define a record structure for a random access file
Parameters:
CHANNEL: The index of the channel
LENGTH: The length of the first field
FIELD$: The string variable to asign it to

Put CHANNEL, RECORD_NUMBER
Output a record to a random access file
Parameters:
CHANNEL: The index of the channel with the open file
RECORD_NUMBER: The number of the record to output

Get CHANNEL, RECORD_NUMBER
Read a record from a random access file
Parameters:
CHANNEL: The index of the channel with the open file
RECORD_NUMBER: The number of the record to read

LOF CHANNEL
Return the length of a "Out", "In" or "Random access" channel
Parameters:
CHANNEL: Index of the channel

EOF CHANNEL
Indicates if the file pointer of a channel is located at the end of the file
Parameters:
CHANNEL: Index of the channel
Value returned:
boolean: True if the file pointer has reached the end, False if not

POF CHANNEL
Set the position of the file pointer in an open channel (type must be "Out" ort "Append")
Parameters:
CHANNEL: Index of the channel

POF CHANNEL
Return the position of the file pointer in an open channel
Parameters:
CHANNEL: Index of the channel
Value returned:
integer: the current position of the file pointer

Close CHANNEL
Close one or all opened files, and in case of output, save the buffered data into it
Parameters:
CHANNEL: Number of the channel (optional)

Parent
Change the current directory to the parent directory

Rename To PATH$, NEWNAME$
Rename a file
Parameters:
PATH$: Path to the file to rename
NEWNAME$: New name of the file

Kill PATH$
Delete a file
Parameters:
PATH$: Path to the file to delete

Open Port CHANNEL, PORT$
TODO! Open a communication port on the machine
Parameters:
CHANNEL: Number of the input channel
PORT$: Name of the port to open

Port CHANNEL
TODO! Return the content of an opened harware port
Parameters:
CHANNEL: Number of the port
Value returned:
integer: Value reported by the port

FSel$ PATH$, DEFAULT$, TITLE1$, TITLE2$
TODO! Open a file selector and return the name of the selected file
Parameters:
PATH$: Path to the directory to display at start
DEFAULT$: Name of the default file to display as selected (optional)
TITLE1$: String to display as title of the selector (optional)
TITLE2$: String to display as secondary title of the selector (optional)
Value returned:
integer: Value reported by the port

Dir First$ PATH$
List the indicated path internally and return the first file in the list
Parameters:
PATH$: Path to the directory to scan, can include * and ? wildcards
Value returned:
string: The name of the first file found

Dir Next$
Return the next file in the list generated by Dir First$
Value returned:
string: The name of the next file found

Exist PATH$
Indicates if a file or directory exists on the disc
Parameters:
PATH$: Path to the file or directory to check
Value returned:

boolean: True if the file or directory exist, False if not

Dir PATH$
List a directory in the current screen
Parameters:
PATH$: Path to the directory to list, can include wildcards * and ?

Set Dir PATH$
Changes the current y
Parameters:
PATH$: Path to the new directory

Disc Info$ PATH$
Return information about the current drive
Parameters:
PATH$: Path to the drive or a directory on the drive (optional)

LDir PATH$
TODO: Output the content of a directory to the printer
Parameters:
PATH$: Path to the drive or a directory on the drive (optional)

Mask Iff MASK
TODO:Indicate what sections of an IFF file to load the next time the Load IFF instruction is used
Parameters:
MASK: Bitmask indicating the load. Example %100: Load palette of picture only, %10000: Load bitmaps only

Command Line$ LINE
Reserved variable. Return or set the parameters of the command that has been used to launch thhe application. For HTML applications, will return the section of the URL after "?"
Parameters:
LINE: The text to use as parameters, separated with commansss inside of the string

Command Line$
Reserved variable. Return or set the parameters of the command that has been used to launch thhe application. For HTML applications, will return the section of the URL after "?"

Set Input CHAR1, CHAR2
Set the end of line characters detected when you input from a random access file
Parameters:
CHAR1: Ascii value of the first character to detect as new line
CHAR2: Ascii value of the first character to detect as new line

Frame Load ... To ... CHANNEL, BANK_OR_ADDRESS, LENGTH
Load frames of an IFF channel into memory
Parameters:
CHANNEL: The index of the IFF channel
BANK_OR_ADDRESS: The number of a memory bank or it's addres
LENGTH: The number of frames to load

Frame Play BANK_OR_ADDRESS, NUMBER, SCREEN

Play IFF frameson screen
Parameters:
BANK_OR_ADDRESS: The number of a memory bank or it's addres
NUMBER: The number of frames to play
SCREEN: The index of the screen to play to, or the current screen if not specified

IFF Anim PATH$, _SCREEN_INDEX, NTIME
Play IFF frameson screen
Parameters:
PATH$: The path to the file to play
_SCREEN_INDEX: The index of the screen to create to play the file
NTIME: The number of times to play, once if ommited

Dir$
Reserved variable. Change current directory

Frame Length CHANNEL, NUMBER_OF_FRAMES
Indicates if a file or directory exists on the disc
Parameters:
CHANNEL: The index of the IFF channel
NUMBER_OF_FRAMES: An eventual number of frames to calculate
Value returned:
integer: The length in bytes of the frames in memory. 0 in this version

Frame Skip BANK_OR_ADDRESS, NUMBER_OF_FRAMES
Skip past an animation frame
Parameters:
BANK_OR_ADDRESS: The index of a memory bank or an address in one
NUMBER_OF_FRAMES: An eventual number of frames to skip
Value returned:
integer: The address of the next frame. 0 in this version

Frame Param
Return a parameter after playing a frame
Value returned:
integer: The amount of time needed to successfully display an animation on screen, measured in 50ths of a second.. 0 in this version