Class TDiscImage

version 1.09

written by Gerald J Holdsworth with help from Jasper Renow-Clarke

Purpose

This is a Delphi class used to load a retro disc image, read the catalogue and output any required files. So far, the following formats are supported:

- Acorn DFS
- Acorn ADFS, floppy formats S, M, L, D, E, E+, F, F+ and Hard Discs
- Commodore 1541
- Commodore 1571
- Commodore 1581
- Commodore AmigaDOS floppy and hard discs

Future additions are planned to be able to write files back to the images.

Methods

Create

This will create an instance of the class, resetting all the properties to their default values.

LoadFromFile(filename: String): TDiscImage

If the filename exists, and the contents are valid, this will read in the catalogue information and populate the variables.

LoadFromStream(F: TStream): TDiscImage

If the contents of the supplied memory stream are valid, this will read in the catalogue information and populate the variables.

ReadFile(filename: String; buffer: TDynByteArray): Boolean

This reads the specified file into buffer. Filename should be the complete path to the file, using '.' as directory separators. This method calls FileExists – detailed below. Will return TRUE on successful read, or FALSE if file not found.

CopyFile(filename, directory: String): Boolean

Copies 'filename' (full path) to 'directory'. Cannot be copied over itself. This calls ExtractFile and WriteFile.
Returns True on success. Currently the supporting methods are not written so this will always return False.

MoveFile(filename, directory: String): Boolean

As CopyFile, except that the original is deleted (calls CopyFile, then DeleteFile).
Returns True on success. Currently the supporting methods are not written so this will always return False.

FileExists(filename: String; Ref: Integer): Boolean

Checks that the given file exists in the current image, where filename is a complete path, using '.' as directory separators. Ref is returned as a reference to where the file is. The top 16 bits are the directory reference, and the bottom 16 bits are the entry reference into directory.

ReadDiscData(address, length: Cardinal; buffer: TByteDynArray): Boolean

Gives direct access to the disc image data, and returns the exact data starting from offset 'address' and continues for 'length' bytes. Reads into 'buffer', which is a TByteDynArray.
Returns TRUE if successful (i.e. address+length is less than the total length of the data), or FALSE if otherwise.

ReadDiscDataToStream(address, length: Cardinal; memory_stream: TStream): Boolean

Gives direct access to the disc image data, and returns the exact data starting from offset 'address' and continues for 'length' bytes. Reads into 'memory_stream', which is a TStream or descendent.
Returns TRUE if successful (i.e. address+length is less than the total length of the data), or FALSE if otherwise.

FileSearch(Search: TDirEntry): TSearchResults

Searches the catalogue for any entries matching the TDirEntry passed. Any fields which are zero or blank are ignored. Results are returned in a TSearchResults, which is a TDirEntry array (not the same as a TDir).

WriteDiscData(addr,side,buffer,count): Boolean

Writes the count bytes of the contents of buffer into the disc image at absolute offset addr. Returns true on success. Fails if it will not fit into the image.

WriteDiscDataFromStream(addr,side,stream): Boolean

Writes the entire contents of the stream into the disc image at absolute offset addr. Returns true on success. Fails if it will not fit into the image.

Public Methods

ResetDirEntry(Entry: TDirEntry)

Resets to blank, or zero, the TDirEntry passed.

Properties

Complete list of all properties used by this class. These are all readable only, currently, except for ProgressUpdate which is writable only. If a property is not used, then it will be zero (Byte/Integer/Cardinal/TDateTime), empty (String/Char) or FALSE (Boolean), unless otherwise stated.

Disc: TDisc

The full catalogue of the disc image. This is an array holding all of the directories. Used by all.

FormatNumber: Byte

Byte representing the format of the image loaded. Used by all.
Top 4 bits - File system:

- 0: DFS
- 1: Acorn ADFS
- 2: Commodore 64/128
- 3: Sinclair Spectrum +3/Amstrad
- 4: Commodore Amiga
- F: none/not recognised (bottom four bits will also be F)

Bottom 4 bits - type of image:

- **DFS:**
  - 0: Acorn Single Sided
  - 1: Acorn Double Sided
  - 2: Watford SSD <future addition>
  - 3: Watford DSD <future addition>
  - 4: Duggan HDFS <future addition>
- **Acorn ADFS:**
  - 0: S (160K, Old Map, Old Directory)
  - 1: M (320K, Old Map, Old Directory)
  - 2: L (640K, Old Map, Old Directory)
  - 3: D (800K, Old Map, New Directory)
  - 4: E (800K, New Map, New Directory)
  - 5: E+ (800K, New Map, Big Directory)
  - 6: F (1.6M, New Map, New Directory)
  - 7: F+ (1.6M, New Map, Big Directory)
  - F: Hard Disc

(for ADFS, use MapType and DirectoryType instead – see below)

Commodore 64/128

0: 1541
1: 1571
2: 1581

Sinclair Spectrum +3/Amstrad

0: DSK <future addition>
1: Extended DSK <future addition>

Commodore Amiga

0: AmigaDOS DD
1: AmigaDOS HD
F: AmigaDOS Hard Disc

Format: String

Textual representation of format of the image. Used by all.

Title: String

Disc title of the disc, if available, otherwise empty. Acorn DFS DS discs will include the titles for both sides here, separated by a space, an ampersand '&', and another space. Used by Acorn DFS; Acorn ADFS; Commodore 1541/1571/1581.

DiscSize: Int64

The size of the image, in bytes. Used by Acorn DFS; Acorn ADFS; Commodore 1541/1571/1581.

FreeSpace: Int64

Calculated free space remaining on disc. Used by all.

DoubleSided: Boolean

Whether disc is reported as double sided (True) or not (False). Used by Acorn DFS; Commodore 1541/1571/1581.

MapType: Byte

Byte representing what type of map:

0x00: ADFS Old Map
0x01: ADFS New Map
0x02: AmigaDOS OFS
0x03: AmigaDOS FFS
0xFF: Not ADFS/AmigaDOS

Used by all.

MapTypeString: String

Textual representation of MapType. Used by all.

DirectoryType: Byte

Byte representing what type of directory:

0x00: ADFS Old Directory
0x01: ADFS New Directory
0x02: ADFS Big Directory
0x10: AmigaDOS normal directory
0x11: AmigaDOS directory cache
0xFF: Not ADFS/AmigaDOS

Used by all.

DirectoryTypeString: String

Textual representation of DirectoryType. Used by all.

DirSep: Char

Character to use to separate the directories in a path. Used by all.

Entries: TDir
The entire catalogue. Used by all.

ProgressLabel: TLabel - write only
If supplied, this will be updated with the progress of the reading of the image. Some images (particularly hard drives) can take some time to read in, and will look as though the application has hung.

TDir structure

Directory: String
The directory name. Used by all.

Broken: Boolean
Whether the directory is considered broken (True) or not (False). Used by Acorn ADFS.

ErrorCode: Byte
An error code to indicate any errors found in the image:

Bit	**Error**
0	Start and End Sequence mis-match (Broken directory)
1	Start and End ID string mis-match or not 'Hugo/Nick' (Broken directory)
2	Start and End ID string not 'SBPr' and 'oven' (Broken directory)
3	Checksum not valid (Broken directory)

A value of 0x00 will indicate no errors. Used by Acorn ADFS.

Title: String
Title of the directory. Used by: Acorn DFS; Acorn ADFS.

Entries: TDirEntry
All the entries under this directory. Used by all.

TDirEntry Structure

Parent: String
Path to file, from root. Used by all.

Filename: String
Filename of entry (no path). Used by all.

Attributes: String
The attributes of the file. Used by: Acorn DFS; Acorn ADFS; AmigaDOS.

Filetype: String
File type of the file. Used by Acorn ADFS D/E/E+/F/F+ when top twelve bits of ExecAddr are set; Commodore 1541/1571/1581.

ShortFileType: String
Short (three character) filetype. In ADFS it is the hex number, while on Commodore it is the three letter abbreviation (e.g. PRG = Program). Used by Acorn ADFS D/E/E+/F/F+ when top twelve bits of ExecAddr are set; Commodore 1541/1571/1581.

LoadAddr: Cardinal
Load address of file. Used by Acorn DFS; Acorn ADFS.

ExecAddr: Cardinal
Execution address of file. Used by Acorn DFS; Acorn ADFS.

Length: Cardinal
Length, in bytes, of file. Used by all.

Timestamp: TDateTime
Date and time stamp of file, as TDateTime. Used by Acorn ADFS D/E/E+/F/F+ when top twelve bits of ExecAddr are set.

Sector: Integer
Start sector of file: Acorn DFS; Commodore 1541/1571/1581; AmigaDOS file.
Sector of disc of location of header: AmigaDOS directory.
Address of location of file: Acorn ADFS S/M/L.
Indirect disc address of file: Acorn ADFS D/E/E+/F/F+.

Side: Cardinal
Which side of the disc file is located. Used by Acorn DFS.

Track: Cardinal
Physical location on disc of start of file. Used by Commodore 1541/1571/1581.

DirRef: Integer
If entry is a directory, reference to catalogue of directory (i.e. reference into TDisc), otherwise -1. Used by Acorn ADFS; AmigaDOS.

EOR: Byte
Reserved

ImageAddress: Cardinal
Reserved

DataType: Cardinal
Reserved

TSearchResults: array of TDirEntry
Results returned from FileSearch.

TFragment Structure

Offset: Cardinal
Offset from the beginning of the map of the fragment.

Length: Cardinal
Length of the fragment.

TFragmentArray: array of TFragment
Complete list of fragments, in order of appearance in the file, for the entire file.

Version history

1.09

- Fixed a bug where ADFS hard drive images without initial 512 byte header are mistakenly described as an ADFS F format
- New methods WriteDiscData and WriteDiscDataFromStream writes data direct to the disc image at a specified address
- Changed the checksum calculation for ADFS header
- WriteFile/WriteFileFromStream now writes DFS files
- ReadDiscData and ReadDiscDataToStream - parameter added to specify side of disc (DFS). Is now ReadDiscData(address,length,side,buffer) and ReadDiscDataToStream(address,length,side,stream)
- Fixed a bug where ExtractFile and ExtractFileToStream does not account for DFS side (see previous entry)

- Rename function 'ReadWord' to 'Read32b' to avoid confusion between Delphi Word (16 bits) and Acorn Word (32 bits)
- Changed output container type of 'Read16b' from Cardinal to Word
- Changed the length variable name from 'len' to 'count' in ReadDiscData and ReadDiscDataToStream functions
- Added new procedures Write32b, Write24b, Write16b and WriteByte to write to the data array
- Property ProgressLabel changed to ProgressUpdate and type changed to TStatusBar
- Methods SaveToFile and SaveToStream implemented
- Fixed a bug in FileExists where searching for a directory would cause it to crash.
- Fixed a bug in ExtractFile where if there are no fragments, for ADFS, it will cause the application to crash.

1.08

- Loads AmigaDOS Kickstart floppies
- Rewritten ExtractFile method to make use of ReadDiscData method, and common code between formats
- Changed ReadBits function to improve the speed
- Combined NewMapChecksum and AmigaGeneralChecksum functions into GeneralChecksum function
- Placeholders put in for new methods: SaveToFile, SaveToStream, Format, WriteFile, WriteFileFromStream
- Combined OldMapChecksum and BootBlockChecksum functions into ByteChecksum function
- Atom DOS root directory changed from ' ' to '$'
- New property - FreeSpace(Int64) : calculated free space remaining on disc
- FreeSpace implemented on DFS and ADFS
- Can read Watford DFS discs
- Found that DFS disc titles can be padded with 0x00, so fixed accordingly
- Found that Watford DFS Master Sequence number is not identical to the lower number, so fixed accordingly
- Placeholders put in for new methods: DeleteFile, RenameFile
- New methods: MoveFile, CopyFile : for moving or copying a file within the image (will only work once WriteFile and DeleteFile are implemented)
- Changed the constant 'dir_sep' to a variable and change it to '/' for AmigaDOS images
- New property 'DirSep' returns the current directory separator
- Updated the ID_Amiga function to account for images without boot block
- Updated the ID_Amiga function to look for root and, hence, account for hard drive images
- Fixed a bug where a Commodore 64 image got IDed as an Old Map ADFS Hard Drive
- Fixed a bug where a failed ADFS ID would clear the variables, even if it had previously been IDed as a DFS
- Calculates the AmigaDOS free space
- Calculates the D64/D71/D81 free space

1.07

- Makes account of a 0x200 byte header that some emulators add to an ADFS hard disc image
- Fixed a bug where there was no checking of number of entries in a directory - ADFS
- LoadFromFile/LoadFromStream now resets all the variables to zero/blank prior to loading a new image
- ReadFile changed to ExtractFile
- New Method: ExtractFileToStream - extracts a file to a memory stream
- New Methods: ReadDiscData and ReadDiscDataToStream - gives direct access to the disc image data
- New functions: ReadByte, to read a byte from the data; and Read16b, to read 2 bytes from the data
- New overloaded functions: ReadWord, Read24b,and Read16b - added extra paramater for big endian
- New data type: TSearchResults - is an array of TDirEntry
- New method: FileSearch: Search for a given file (details as TDirEntry), returns a TSearchResults
- Moved ResetDirEntry from private to global and is now available outside of the class
- Property 'Disc' is now read only

- Changed all accesses to the data to using ReadByte or Read16b
- Changed FileExists to accept a Cardinal as second parameter, instead of Integer
- Extended the sub-format strings from 7 to 15 in FormatString
- Added check for boot block checksum for ADFS multi-zone images
- New property: 'ProgressLabel', is a TLabel which is to give the user feedback on progress on the reading of the image
- Can now load AmigaDOS floppy discs (OFS only)
- Extended MapType to include AmigaDOS FS types. MapTypeToString updated accordingly.
- Extended DirectoryType to include AmigaDOS DIRC. DirectoryTypeToString updated accordingly.

1.06

- ResetDirEntry never reset the ShortFileType property
- Changed the ampersand (&) to 'and' on dual sided DFS discs
- Directory name and filename had control characters left in - added new procedure to remove them
- Added new function 'LoadFromFile' to load an image.
- Constructor 'Create' now just creates the instance and clears all the variables.
- The Title and DiscSize properties are populated for Commodore D64/D71/D81 images
- New property 'TDir.ErrorCode' - for if an error is detected.
- New method 'LoadFromStream' - loads an image from a memory stream
- Changed 'LoadFromFile' to use LoadFromStream

1.05

- Changed private variable names to have 'F' prefix
- Changed 'Format' property to FormatString
- Expanded 'SSD' and 'DSD', in DFS strings, to 'Acorn' and 'Watford' SSD and DSDs
- Added FormatNumber property (see below)
- IDs a DFS disc by contents and not file extension
- IDs an ADFS disc by contents and not file extension
- New properties: MapType - ADFS map type; DirectoryType - ADFS directory type
- IDs a Commodore 1541/1571/1581 by contents and not file extension
- DoubleSided property gets set/reset with Commodore images
- Expanded the Commodore file types from three characters to full names
- New property in TDirEntry : ShortFileType - the three letter filetype (hex number for ADFS)
- Added skeleton functions to ID a Sinclair and Amiga image
- Added skeleton functions to read a Sinclair and Amiga image
- New properties: MapTypeString and DirectoryTypeString - returns string versions of MapType and DirectoryType
- Moved filetype array constants to appropriate functions, instead of being globally available

1.04

- Checks for broken directory in ADFS
- Re-structured the layout of the code so that procedures/functions for each format are together
- Re-written ReadADFSDir function
- Added ReadWord and Read24b functions to read in 32 and 24 bits
- Reads in ADFS E+ and ADFS F+ format
- Included directory checks (looks for Hugo/Nick/SBPr/oven; checks the Sequence Numbers match and calculates the DirChkByte)
- Added ROR13 function to ROtate Right a Cardinal by 13 bits
- Added property to TDir - 'Broken' - indicates whether a directory is broken (i.e. has failed one of these checks)
- Removed DirectorySeperator as it was spelt wrong, and was not required
- Renamed a lot of the procedures/functions to use uppercase then lowercase
- Reads in the header information on Old Map discs
- Runs the checksum on the old map headers
- Added new property - Title - for the disc title

- Added new property - DiscSize - for the size of the disc in bytes
- Changed ReadString to accept an optional third parameter whether to stop on control characters (default is True)
- Added RemoveSpaces procedure to remove trailing spaces from a string
- Runs the checksums on zones for new maps
- Now takes account of the fact that the first fragment of an object may appear after later fragments (i.e. adjusts the zone to start searching from)
- Removed DSD property from TDirEntry and moved to TDiscImage as DoubleSided
- Added disc size and disc titles to DFS discs
- Fixed a bug where FileExists could not find a find, with a directory prefix, on a DFS disc

1.03

- Rewritten ConvertDxxTS
- Added new property - 'DirectorySeperator' - defines the separator between directories: default is '.'
- The DiscAddrToOffset function now makes proper use of values from the disc record to calculate offsets
- New function isbitset(v,b) to determine if bit 'b' is set in value 'v'
- Identifies filetype, if available, on Acorn ADFS
- New function ConvertTimeDate to convert from RISC OS timestamp to Delphi TDateTime
- New property Timestamp, for Acorn ADFS, where files are timestamped

1.02

- Revised the method of calculating disc address from fragment ID and sector
- Changed the method of reading in fragment id's to account for idlens other than 15
- Fixed a bug where a file won't get downloaded from ADFS if it is not in the root directory (readFile)
- Can deal with an ADFS F format

1.01

- Detection of Amiga adf images, and separation from ADFS images
- Detection of Sinclair/Amstrad dsk images
- readFile no longer looks for file when there is no valid disc image
- Detection of Commodore 8050 and 8250 images
- readFile is no longer case insensitive
- New method: FileExists looks for a file in the image
- Parent attribute is now the complete path to the file, minus the filename
- Bugs fixed in readFile:
    - Read an extra byte per fragment with Commodore files
    - Failed to find files on DFS format
    - Failed to correctly locate the data for DFS and Old map ADFS files

1.00

- Initial version (converted to class from Delphi unit)