Acorn DFS disc format

beebwiki.mdfs.net/Acorn_DFS_disc_format

Acorn DFS is a simple filing system developed by Acorn Computer for its 8-bit microcomputer range. It provides storage for up to 31 files and 200 kilobytes of data on a single sided disc, and up to 124 files and 800 kilobytes on a fully equipped BBC Micro with two double sided disc drives.

History

The Acorn DFS (Disc Filing System) format was introduced on the System series of modular computers in the late 1970s. The disc interface cards were based on the Intel 8271 floppy disc controller (FDC).

In 1981 Acorn designed the interface into the Proton (later to become the BBC Micro) virtually unchanged. This was a significant mis-step for Acorn as the 8271 fell into short supply within two years, superseded by double density controllers. A number of third party vendors stepped in offering interface boards to fit the 8271 socket; for the most part these were compatible with Acorn DFS but also offered their proprietary double density formats.

Acorn made good in 1985, installing the WD1770 controller in the Model B+, but did not extend the DFS to take advantage of the extra storage capacity. In the meantime, their Plus 3 expansion pack for the Electron had incorporated the new 3½ inch disc drive and, exclusively, the next generation ADFS which used double density to the full. Both DFS and ADFS were built into the Master series of computers before the Master Compact and Archimedes series committed to ADFS.

Physical format

Acorn DFS is a single sided, single density 40 or 80 track format. A double sided disc can hold two DFS volumes. Each track carries ten 256 byte sectors numbered from 0 to 9.

The standard physical media is a 5¼ inch floppy disc, as this was the only reasonable option in 1981. The 'education friendly' 3-inch and 3½ inch formats, though marketed to BBC users in the early 1980s, did not catch on as DFS media. But since ADFS machines and PCs embraced the latter, it has had a clandestine and continuing user base.

As the format was first implemented on the Intel 8271 floppy disc controller, it is a variant of the IBM 3740 diskette format, and so much of the terminology has been carried across. For further and lower-level details please see the Intel 8271 datasheet.

Sector IDs

Each sector is given an ID at format time; the parameters of the ID, listed below, are used by the DFS and the floppy disc controller to address sectors.

- *C*, the *cylinder number* ranges from 0 on the outermost track to 39 or 79 on the innermost track.
- *H*, the *head number* = 0. Acorn DFS ignores this after formatting. Some DFS formatters set it to 1 on the upper surface of the disc (corresponding to drives 2 or 3.)
- *R*, the *record number* runs sequentially from 0 to 9 within each track. There is a standard *track skew* of 3; that is, the sectors having *R* = 0 are the first sector of the first track, the fourth sector of the second track, the seventh of the third, the tenth of the fourth, and so on.
- *N*, the *record length* = 1, showing that the sector length is $2^{7 + 1}$, or 256 bytes.

All these parameters can potentially take any value between 0 and 255, and so confuse Acorn DFS and some non-standard controllers. Some copy protection schemes make use of this feature.

Gap lengths

The IBM 3740 format specifies five kinds of *gap* that appear at the beginning and end of each track, between sectors and within sectors, to allow the floppy disc hardware time to settle and prepare. When formatting, the 8271 takes parameters specifying the length of three of these gaps. For detailed information on the gaps and their locations, please see the Intel 8271 datasheet.

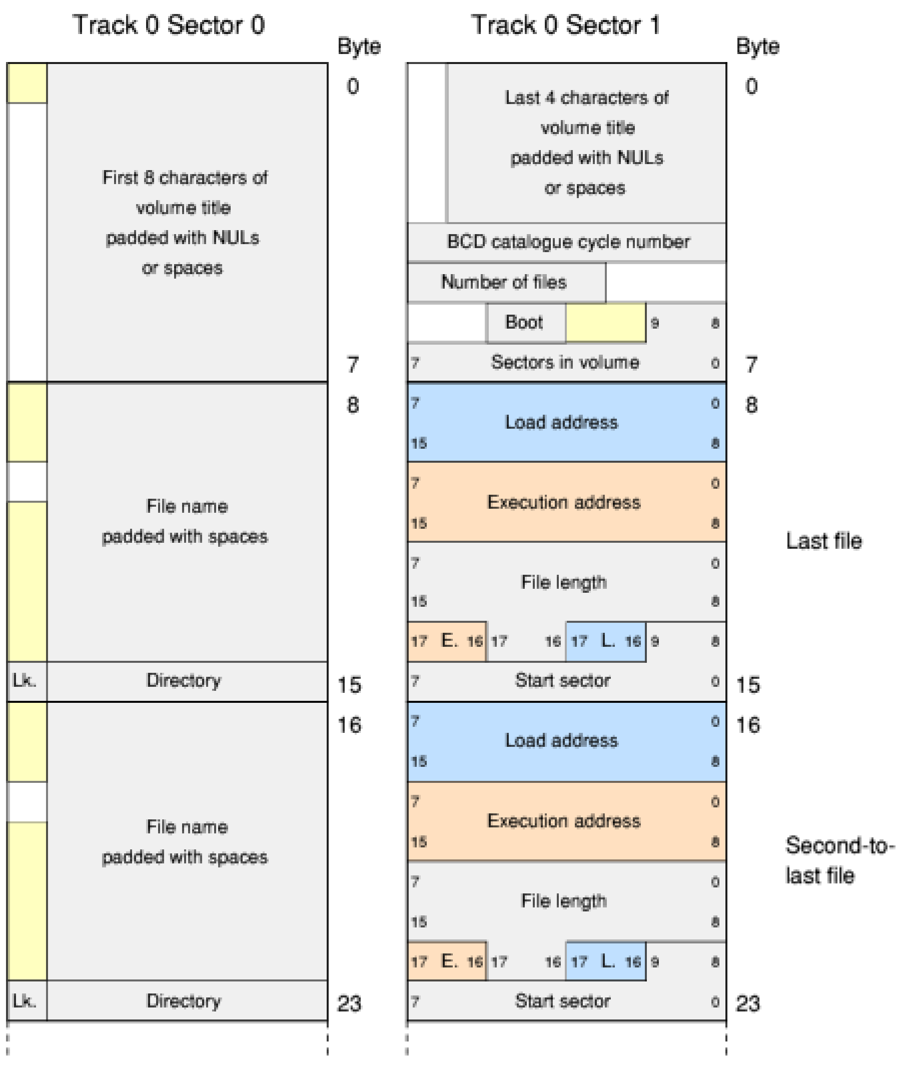
In practice the gap parameters can vary within quite a wide range and still appear to work and some third-party formatters tweek the gap sizes to improve on the standard Acorn formatter. The `*FORM40` and `*FORM80` utilities, on the Utility disc, format using command `&63` and parameters `<track number>, &10, &2A, &00, &10`. That is, they specify gap3 = gap1 = &10, to write 16 bytes of &FF plus 6 bytes of &00; and gap5 = 0, which eliminates the pre-index gap and the index address mark.

Some third party DFSs format with gap1 and gap3 set to &12.

Catalogue format

Before a disc can be used reliably by Acorn DFS, it must be formatted and a blank catalogue written to the first two sectors. This is done automatically by the `*FORM40` and `*FORM80` utilities and third party equivalents. To create a blank catalogue a program need only clear the two sectors and then initialise the disc size field.

The fields are packed in the catalogue according to the following diagram:


Track 0 Sector 0
Byte
First 8 characters of volume title padded with NULs or spaces
File name padded with spaces
Lk.
Directory
Track 0 Sector 1
Last 4 characters of volume title padded with NULs or spaces
BCD catalogue cycle number
Number of files
Boot
Sectors in volume
Load address
Execution address
File length
Start sector
Last file
Second-to-last file

Disc descriptor fields

Disc title	Up to 12 printable ASCII characters, padded with NULs (&00) or spaces. The first eight characters are placed in sector 0 bytes 0 to 7, the last four characters in sector 1 bytes 0 to 3.
Cycle number	A binary-coded decimal (BCD) value where each hex nibble takes a value from 0 to 9. This starts at &00 and is incremented each time the catalogue is rewritten. Provides a simple version control system and allows applications to confirm that their internal copies of the catalogue are up to date. The cycle number is stored in sector 1 byte 4.
File offset	The offset to the last valid file entry in each sector of the catalogue. It is therefore 8 times the number of files. There may be up to 31 files on the disc. The file offset is stored in sector 1 byte 5.

Boot option	Selects the action to be taken when the disc is booted. The field uses two bits of byte 6 in sector 1:

Bit 5	Bit 4	Action
0	0	No action
0	1	`*LOAD $.!BOOT`
1	0	`*RUN $.!BOOT`
1	1	`*EXEC $.!BOOT`

Disc size	The total number of sectors on this side of the disc. This is a 10-bit field with the low 8 bits in sector 1 byte 7, and the others in the low two bits of sector 1 byte 6. The unused bits of the latter (bits 2, 3, 6 and 7) must be cleared.

Disc title padding

During `*TITLE`, Acorn DFS versions 1.00 and 1.20 pad the disc title with spaces; versions 0.90, 0.98, 2.00 and above pad it with NULs. Either choice is correct, although NULs print more neatly and are compatible with firmware that assumes old DFS behaviour.

File descriptor fields

There is room for 31 file entries in the catalogue. Each entry takes up eight bytes in each of the two catalogue sectors. In the descriptions below any multiple of 8 up to 240 can be added to the byte offsets.

Files must be listed in descending order of start sector with no gaps in the catalogue. Empty files should be given a start sector of 2. Files cannot be fragmented but there may be free sectors between files.

File name	One to seven valid file name characters stored in sector 0 bytes 8 to 14. Valid characters are the printable ASCII characters between &20 and &7E inclusive, except . : " # * and space. The field is padded with spaces. When opening a file the DFS searches for it by name.
Directory	One valid file name character stored in the low 7 bits of sector 0 byte 15. Identifies the *directory* (namespace) to which the file belongs. DFS directories are like those in ADFS, DOS or Linux except they cannot nest (they are *all* root directories) and they share the volume catalogue. The combination of file name and directory must be unique in the volume.
Attribute	If the top bit of sector 0 byte 15 is set, then the file is locked and may not be altered or deleted.

Load address	An 18 bit address in memory where the file should be `*LOAD`ed by default. The low 8 bits are in sector 1 byte 8; the next 8 bits in sector 1 byte 9; the top two bits are in bits 2 and 3 of sector 1 byte 14. If the file is not meant for `*LOAD`ing (e.g. a sequential file or text) all 18 bits should be set.
Execution address	An 18 bit address in memory to be jumped to when the file is `*RUN`. This need not be within the bounds of the loaded file. The low 8 bits are in sector 1 byte 10; the next 8 bits in sector 1 byte 11; the top two bits are in bits 6 and 7 of sector 1 byte 14. If the file is a sequential file or text then all 18 bits should be set.
File length	An 18 bit value giving the number of bytes in the file. The low 8 bits are in sector 1 byte 12; the next 8 bits in sector 1 byte 13; the top two bits are in bits 4 and 5 of sector 1 byte 14.
Start sector	The 10 bit *logical block address* of the first sector that contains the file. The low 8 bits are in sector 1 byte 15; the top two bits are in bits 0 and 1 of sector 1 byte 14. The cylinder and record numbers are the result and remainder, respectively, when the LBA is divided by 10. A start sector of 0 or 1 is invalid as the file would overlap the catalogue.

Note on load and execution addresses: If bits 16 and 17 of the address are both set, the address refers to I/O processor memory and OSFILE will return the address ORed with &FFFF0000 in its parameter block. Otherwise it is an address in second processor memory and OSFILE will return the address as is. In practice the bits are sometimes found clear when they should be set.

Identifying and validating a DFS disc image

Unlike its successor ADFS, the DFS format contains no self-identification strings or checksums, so recognising a disc or disc image as Acorn DFS is a complex and unreliable process.

If one has a floppy disc in hand it is often enough to confirm that the physical format is the same (see above) as Acorn DFS is the majority producer. On the other hand DFS disc image files are normally given the extension `.ssd` or `.dsd`, indicating a single or double sided disc image respectively.

Often the task is to identify which format variant a disc or image contains. This is more a matter of finding valid catalogues in the appropriate locations.

To check that a pair of sectors contains a valid standard DFS catalogue

- Firstly confirm that all bits marked white or cream in the above diagram are zeroes; if not, the catalogue can be rejected immediately.
- Check also that the file offset is a multiple of 8.
- The disc size must be between 2 and 800 sectors inclusive, although the format allows up to 1023 sectors, and disk images may contain up to this many.
- The disc title consists of printable ASCII characters, padded with NULs or spaces.

- For each file entry in use:
  - The file name field must contain one to seven valid characters and be padded with spaces. Valid characters are the printable ASCII characters between &20 and &7E inclusive, except . : " # * and space.
  - The directory character (without the attribute bit) must be a valid file name character.
  - The file name (taking the directory character into account) must be unique.
  - The start sector must be more than 1 and less than the disc size.
  - Skipping over zero-length files:
    - The start sectors must run in strictly descending order.
    - No file may overlap the one before it in the catalogue. That is, *start_sector_2* + ((*length_2* + 255) DIV 256) ≤ *start_sector_1*.
    - No file may overshoot the end of the disc. That is, *start_sector* + ((*length* + 255) DIV 256) ≤ *disc_size*. Equivalently, the first file in the catalogue must not overlap the imaginary file whose start sector field is the disc size field.

Further heuristics

Any catalogue complying with the above is a valid DFS catalogue; however, only a tiny subset of these look like catalogues seen in real life. As an informal test a pair of sectors that match many of the following is more likely to be a catalogue:

- The disc size is 400 sectors (40 tracks) or 800 sectors (80 tracks.)
- There is a file named `!BOOT` in directory `$`.
- The 16th and 17th bits of the load and execution (exec) addresses are all clear or all set in each file entry.
- There are few directories, or a few unique names in several directories.
- Most directory characters are `$` (the default.)
- Some files have load addresses of the form &3nn00 (nn=0E, 11, 19, 1B, 1D, 1F or 21) and exec addresses of &3801F or &38023. (BASIC programs.)
- Some files have load and exec addresses the same, usually &xx00 or &3xx00, and often have length &yy00. (Screen dumps and machine code images.)
- Some files have load and exec addresses of 0, &30000 or &3FFFF (Text and sequential data.)
- Many files are less than 2K; few files more than 20K; no files more than 64K except text and sequential data.
- The file names have very few lowercase letters.
- The file names contain commonly-occurring pairs of letters.

Variants

Acorn System DFS

Discs for the System series have an identical catalogue format to those for the BBC Micro, except all entries are in a directory " " (space), disks do not have a cycle number, and files do not have bits 16 and 17 of load and execution addresses[1]. The places they would occupy are reserved and must be cleared. A BASIC program named `DCONV`, on the Utility disc, performs this task.

Watford 62 DFS

This system places a second catalogue in sectors 2 and 3 of track 0, allowing 62 files on one volume. To identify the second catalogue it sets its title to 12 × &AA bytes.

Without due care a system running Acorn DFS can corrupt a Watford disc as it is unaware of the extra files. On such a system the user may run a `*SWAP` utility to exchange the catalogues. In practice there is a special entry in the first catalogue (directory `!`, name `!!!!!!!`, length &200, start sector 2) protecting the second one from being overwritten.

If any file in the first catalogue starts before sector 4 and it is not the special file, the disc is not a Watford DFS disc.

Opus DDOS

This is a double density format, but it is mentioned here as the volume catalogues are based on Acorn DFS.

DDOS discs are partitioned into between one and eight volumes. Track 0 is reserved for the DFS-style volume catalogues, plus a disc catalogue unique to DDOS.

In each volume catalogue:

- The *start sector* may be any number less than the disc size, as DDOS adds a partition offset and so the catalogue is not in the same space as the files.
- The *disc size* is the size of the current volume. It is an 11 bit quantity, the top bit being in bit 2 of sector 1 byte 6.
  If and only if this bit is set:
  - The *file length* steals an extra bit from the execution address of each file (bit 6 of sector 1 byte 14+n.)
  - The *start sector* steals an extra bit from the load address of each file (bit 2 of sector 1 byte 14+n.)

Watford DDFS

This is also a double density format with catalogues similar to Acorn DFS. It uses 19 bit file lengths and 11 bit LBAs throughout, but the 'extra' bits are stored in the top bits of the disc title and file names, which are cleared in Acorn DFS.

Applications that handle the Acorn DFS format

On physical discs

- *DFS Xfer*, part of **DFS Explorer** by Jon Welch, runs between a BBC Micro fitted with DFS and a PC, and transfers images of physical discs over a serial link.
- **Xfer in C**, by Jon Welch and others, is the CLI based predecessor of DFS Xfer and also transfers individual files.
- **Omnidisk / Omniflop**, by Sherlock Consulting, reads and writes several native formats, including Acorn DFS, on compatible PCs.
- **DiskToImg and ImgToDisk**, runs on BBC/Master or RISC OS.
- **Anadisk**, by Sydex, Inc. is an older shareware utility to produce a disc image from a physical disc on compatible PCs.
- Plenty of **utility ROMs**, **magazine listings** and **PD programs** for the BBC Micro have built-in support for the DFS catalogue; primarily because Acorn DFS's API does not offer a machine readable catalogue of *all* files on the drive.

In disc images

- **DFS Explorer**, by Jon Welch, is a GUI-driven Windows application to manipulate DFS disc images including those transferred by DFS Xfer.
- **Omnidisk / Omniflop**, by Sherlock Consulting, will also display, add and extract files from DFS disc images.
- **bbcim**, by W.H.Scholten, is a versatile, CLI-based Linux tool to manipulate many types of BBC disc image.
- **MkImg and UnImg**, runs on BBC/Master, RISC OS or Windows.

External links

Summary of WDFS and HDFS catalogue format extensions
Watford DFS manual
Intel 8271 datasheet

References

1. ↑ `DCONV`, lines 190–210.

-- beardo 08:25, 15 November 2006 (GMT)