DiskCopy 4.2 format specification

From 68kMLA Wiki

This article describes the **file format for Apple Disk Copy 4.2** .image files

Much information comes from the CiderPress and Mini vMac source codes. More authoritative information comes from nulib.com (http://www.nulib.com/library/FTN.e00005.htm). Some info on tags from Inside Macintosh, 1st ed. page II-212 (http://www.pagetable.com/?p=50). This format is also used in DiskCopy 4.1. DiskCopy 6.3.3 uses a variant with tags omitted. DART is a variant which adds compression.

Contents

Resource fork notes

Disk Copy 4.2 files have a resource fork, but it only contains a copy of the data and tag checksums and can be safely ignored; files without the fork will still work perfectly.

- Disk copy 4.2 images have a type of 'dImg' and creator of 'dCpy', without these the program will not recognize the file; they can be easily added to images missing them with resedit.

DC42 File format Overview

offset	type/size	contents
0x00	byte	Length of image name string ('Pascal name length')

0x01-0x3F	63 bytes	Image name, in ascii, padded with NULs
0x40-0x43	BE_UINT32	Data size in bytes (of block starting at 0x54)
0x44-0x47	BE_UINT32	Tag size in bytes (of block starting after end of Data block)
0x48-0x4B	BE_UINT32	Data Checksum
0x4C-0x4F	BE_UINT32	Tag Checksum
0x50	byte	Disk encoding
0x51	byte	Format Byte
0x52-0x53	BE_UINT16	'0x01 0x00' ('Private Word') AKA Magic Number
0x54-...	variable	Image data
...-EOF	variable	Tag data

Specifics of data fork sections

0x00: Length of image name string

Technically this is part of the 0x01-0x3f area, as pascal strings apparently store their length as their first byte. Effectively the address of the last non-NUL byte of the image name. bytes after the end address are ignored, and sometimes are used to hold extra information, or hold garbage (in the case of the 1.44mb system 6.0.8 startup disk, this is leftovers in memory from the System Additions disk, so the string ends up being System Startupns)

- Note: A special (bug) case happens when the disk name is "-not a Macintosh disk", which the name is set to when using dc42 on non-mac diskettes. In that case, the length is set one byte too high; this is probably a bug in dc42 that was just never fixed.

0x01-0x3F: Image name

This is the image name string. It is copied from the volume name of the disk or diskette being imaged.

0x40-0x43: Data block size in bytes

This has one of 4 values on most diskettes:

```
00 06 40 00 (409600 bytes) for 400k GCR disks
00 0c 80 00 (819200 bytes) for 800k GCR disks
00 0b 40 00 (737280 bytes) for 720k MFM disks
00 16 80 00 (1474560 bytes) for 1440k MFM disks
```

0x44-0x47: Tag size in bytes

This is typically 12 bytes for every 512 bytes in the image, apparently stored in the data mark in each sector on

the media. The format for Tag data is described at the bottom of the document, and is important for repairing damaged disks using disk doctor. It is also very important on Lisa Diskettes (according to LisaEM source).

```
00 00 25 80 (9600 bytes) for 400k disks w/tags
00 00 4b 00 (19200 bytes) for 800k disks w/tags
00 00 00 00 for diskettes with no tags
```

0x48-0x4B: Data Checksum

The algorithm for this is: start with 00000000, add each consecutive 16 bit Big Endian word of the section, then rotate the 32 bit result right by 1 bit.

0x4c-0x4F: Tag Checksum

The algorithm for this is the same as for the data, however the first 12 bytes of the tag section, if present, are skipped (probably due to a bug in an older disk copy version and kept for compatibility).

- Tag Checksum is 00 00 00 00 if no tag data is present.

0x50: Disk encoding

This byte describes the encoding used for the disk the data was imaged from, from a 'what type of disk is this?' perspective.

```
00 = GCR CLV ssdd (400k)
01 = GCR CLV dsdd (800k)
02 = MFM CAV dsdd (720k)
03 = MFM CAV dshd (1440k)
Other encodings may exist, as DC42 was originally designed to be able to image HD20 disks.
```

0x51: Format Byte

This byte has one of two meanings, depending on whether the disk is GCR format 400k or 800k, or MFM format. The byte is completely ignored for the rare GCR-on-HD format (which always has a 1:1 interleave and is always 2 sided).

- If disk is GCR format 400k or 800k:

```
This byte is a copy of the GCR format nybble (6 bits),
which appears in the headers of every GCR sector.
$02 = Mac 400k
$12 = (documentation error claims this is for mac 400k disks, but this is wrong)
$22 = Disk formatted as Mac 800k
$24 = Disk formatted as Prodos 800k (AppleIIgs format)
$96 = INVALID (Disk was misformatted or had GCR 0-fill (0x96 which represents data of 0x00)
written to its format byte)
 Values for bitfield:
 76543210
 ||||||||
 |||\\\\\- These 5 bits are sector interleave factor:
 |||           setting of 02 means 2:1 interleave: 0  8  1 9  2 10 3 11 4 12 5  13 6  14 7  15
 |||           setting of 04 means 4:1 interleave: 0  4  8 12 1 5  9 13 2 6  10 14 3  7  11 15
 ||\------ This bit indicates whether a disk is 2 sided or not. 0 = 1 sided, 1 = 2 sided.
 \\------- always 0, as GCR nybbles are only 6 bits
```

- If disk is MFM format:

```
This byte is used to define MFM sector size and whether the disk is
two sided or not.
Interleave is ALWAYS 1:1 for these formats.
$22 = double-sided MFM diskettes with 512 byte sectors
 Values for bitfield:
 76543210
 ||||||||
 |||\\\\\- These 5 bits are sector size as a multiple of 256 bytes
 |||          i.e. 02 = 2*256 = 512 bytes per sector
 ||\------ This bit indicates whether a disk is 2 sided or not. 0 = 1 sided, 1 = 2 sided.
 \\------- unused, always 0
```

0x52-0x53: Private Word/Magic Number

This is more or less the 'magic number' of any DC42 format file; if it is not $01 $00, it is not a DC42 format file.

0x54-...: Image data

The size in bytes of the data stored here is the value at 0x40-0x43.

...-EOF: Tag data

The size in bytes of the data stored here is the value at 0x44-0x47.

Tag Data format

The tag data is 12 bytes per 512-byte disk sector, and is stored, like the Image data, in sector order. The actual format for each 12-byte block of the Tag data differs for Lisa, MFS and HFS disks, and for MFS or HFS any of them may be wrong or absent! be warned!

- The Tag format for Lisa 400k or 800k disks is currently unknown, but without tags the disks will not function.
- For MFS filesystems the Tag format is as follows:

```
BE WARNED: when reading tag data, if the bit at 00 40 00 00 of any of the 3 32 bit words of the tag is set, t
 data for the sector it is part of is trashed and can be ignored. There IS a puprose to the data written when
offset       type/size      contents
0x00      BE_UINT32     File number on disk, within MFS filesystem
0x04      BE_UINT16     Flags bitfield:
          FEDCBA98 76543210
          |||||||| ||||||||
          |||||||| |||\\\\\- unknown, seems unused
          |||||||| ||\------ If set, Tag for this sector is not valid.
          |||||||| \\------- unknown
          |||||||\---------- sector content type: 0: system file; 1: user file (guessed)
          ||||||\----------- sector is part of a: 0: data fork; 1: resource fork
          |\\\\\------------ unknown
          \----------------- unknown, sometimes set on the last few sectors of a data or resource fork
0x06      BE_UINT16     Logical block number within the file
0x08      BE_UINT32     Time of last modification, in seconds since 0:00:00, 1/1/1904
Note that the last mod time may be different on the final sector of a file; this may indicate something speci
```

Retrieved from "https://wiki.68kmla.org/index.php?title=DiskCopy_4.2_format_specification&oldid=5278"
Category: Disk imaging

- This page was last modified on 11 November 2010, at 16:58.
- Content is available under GNU FDL unless otherwise noted.