AARU

AaruFormat Specification

Version 1.0

8th May 2022

AaruFormat Specification

Version History

Date	Version	Branch	Author	Modifications
08 May 2022	1.0	Official	Natalia Portillo	Initial version

Table of Contents

1. Introduction

This document is the detailed specification of AaruFormat.

Audience

This specification is directed to emulator developers, software preservators, archives, museums and collectors, that want to have a common file format where to store, archive and manage, dumps and copies of any type of computer storage.

Scope

The scope of this specification is to define an open, free and universal file format able to store and describe any kind of digital or analog storage media for computer systems, in a clear and extensible way that allows for new media to be easily added, along with any kind of metadata describing them, plus verification and recovery data.

Currently the idea is for it to be able to store punch cards, disks (magnetic, optical, magnetoptical) and tapes (analog and digital tapes), decoded or as audio tones and as magnetic or optical fluxes, with any kind of copy protection or absence of it.

Because of its design goals, the format here described may not be the best for reproduction or emulation, but it pretends to be the best for archival and preservation.

There are other formats pretending to achieve some of these goals, and precisely that's why this format is designed. To be a single, universal, extensible, standard, eliminating the need to use a different format for each type of storage.

2. Definitions

Types

All binary types used in this specification are stored as little-endian values on the file. This specification follows the C syntax to denote hexadecimal values, and requires the read some knowledge on programming.

Endianness

Unless specified otherwise for any reason all fields in this specification are considered to be in Little-Endian format, that is the hexadecimal number 0x12345678 is stored in disk as the following sequence of bytes: 0x78 0x56 0x34 0x12.

Header identifiers

Header identifiers are 4 ASCII characteres stored as a sequence of bytes inside a single 32 bits pack. They are shown in this specification enclosed in single quotes. For example, the header identifier 'AARU' should be stored on disk as 0x41 0x41 0x52 0x55.

Integers

Integer values are designated in this specification started with a single caps leter indictating if signed or unsigned (S or U), continuing with Int and ending with the number of bits able to be stored in them.

That so, the signed integers should be: *SInt8*, *SInt16*, *SInt32*, *Sint64* and *SInt128*.

And the unsigned integers should be: *UInt8*, *UInt16*, *UInt32*, *UInt64*, *UInt128*.

Strings

All strings are stored as a sequence of bytes, in Unicode's UTF-16 little endian encoding and terminated and filled with NULL (0x00) bytes.

String8 values mean the string is stored in Unicode's UTF-8 encoding and terminated and filled with NULL (0x00) bytes.

StringA values mean the string is stored in ASCII encoding and terminated and filled with NULL (0x00) bytes.

Timestamp

All timestamps used in this specification are stored as a signed 64bit integer (SInt64) counting the number of nanoseconds in the UTC timezone after/before the epoch of 1st January 1601 at 00:00 of the Gregorian Calendar. This epoch is chosen because it is when the leap-year scheme was adopted.

Media tag

A media tag is a piece of data that is physically present in the media but it's not part of the user data. It can be the table of contents, some manufacturing information, sector replacement tables, etc.

Sector tag

A sector tag is a piece of data that is physically present in the media, once per each sector, but it's not part of the user data. It can be addressing information, error detection or correction ifnormation, encryption metadata, etc.

NULL

NULLs are 0x00 bytes.

3. The Header

The master header is the most important part of an AaruFormat file. It describes the size and version of the file, as well as the contents and file-wide metadata.

Type	Size	Name	Description
UInt64	8 bytes	identifier	The header identifier, always 'AARUFRMT'. Old images can be found with ‘DICMFRMT’ but new ones must never be created with this value.
String	32 bytes	application	The name of the application that created the file.
UInt8	1 byte	imageMajorVersion	Major version of AaruFormat. This specification version is 2.
UInt8	1 byte	imageMinorVersion	Minor version of AaruFormat. This specification version is 0.
UInt8	1 byte	applicationMajorVersion	The major version of the application that created this file.
UInt8	1 byte	applicationMinorVersion	The minor version of the application that created this file.
MediaType	4 bytes	mediaType	Type of media contained in this image. See Annex A.
UInt64	8 bytes	indexOffset	Offset in bytes to the index header currently in use by this image.
SInt64	8 bytes	creationTime	Creation time of this file.
SInt64	8 bytes	lastWrittenTime	Last modification time of this file.

4. The Blocks

The blocks in AaruFormat are the construction pieces of the file, that contains the data and metadata from the media represented by it.

*The index block ('INDX') *DEPRECATED**

The index block contains pointers to all the blocks present on the file. It consists in the header followed by the number of entries indicated by the header.

There can be several index blocks in a file, representing points in the past of the file, but only the last one shall be the one pointed by the main header.

This block is deprecated and new images should not contain it.

Type	Size	Name	Description
UInt32	4 bytes	identifier	The index block identifier, always 'INDX'
UInt16	2 bytes	entries	The number of entries following this header
UInt64	8 bytes	crc64	CRC64-ECMA checksum of the entries following this header

Index entries

Type	Size	Name	Description
UInt32	4 bytes	blockType	The type of block this entry points to.
UInt32	4 bytes	dataType	The type of data the block pointed by this entry contains.
UInt64	8 bytes	offset	The offset in bytes from the start of the file where the block pointed by this entry starts.

The index block version 2 ('IDX2')

The index block contains pointers to all the blocks present on the file. It consists in the header followed by the number of entries indicated by the header.

There can be several index blocks in a file, representing points in the past of the file, but only the last one shall be the one pointed by the main header.

Type	Size	Name	Description
UInt32	4 bytes	identifier	The index block identifier, always 'INDX'
UInt64	8 bytes	entries	The number of entries following this header
UInt64	8 bytes	crc64	CRC64-ECMA checksum of the entries following this header

Index entries

Type	Size	Name	Description
UInt32	4 bytes	blockType	The type of block this entry points to.
UInt32	4 bytes	dataType	The type of data the block pointed by this entry contains.
UInt64	8 bytes	offset	The offset in bytes from the start of the file where the block pointed by this entry starts.

The data block ('DBLK')

The data block contains data from the media.

It consists of a header followed by the compressed, or uncompressed, data.

Type	Size	Name	Description
UInt32	4 bytes	identifier	The data block identifier, always 'DBLK'
UInt16	2 bytes	type	The data type contained in this block. See Annex B.
UInt16	2 bytes	compression	The compression algorithm used in the data. See Annex C.
UInt32	4 bytes	sectorSize	The size in bytes of the sectors contained in this data block if applicable.
UInt32	4 bytes	cmpLength	The size in bytes of the compressed data that follows this header.
UInt32	4 bytes	length	The size in bytes of the data block when decompressed.
UInt64	8 bytes	cmpCrc64	The CRC64-ECMA checksum of the compressed data that follows this header.
UInt64	8 bytes	crc64	The CRC64-ECMA checksum of the decompressed data.

A data block can contain user data, that is, media sectors, or separate data, for example media or sector tags.

In case a single data block contains several items (e.g. sectors or sector tags), the *sectorSize* frield indicates the size in bytes of each individual item. In case it contains a single item (e.g. media tags), *sectorSize* must be set to 0.

The deduplication table ('DDT')*

The deduplication table is an array of pointers laid out sequentially. Each sector on the media represents a pointer in the deduplication table. At least one deduplication table with a data type of *UserData* must be present in the image.

Type	Size	Name	Description
UInt32	4 bytes	identifier	The deduplication table identifier, always 'DDT*'
UInt16	2 bytes	type	The data type pointed by this table. See Annex B.
UInt16	2 bytes	compression	The compression algorithm used in the table. See Annex C.
UInt8	1 byte	shift	The shift used to calculate the position of a sectorin a data block pointed by this table.
UInt64	8 bytes	entries	How many pointers follow this header.
UInt32	4 bytes	cmpLength	The size in bytes of the compressed table that follows this header.
UInt32	4 bytes	length	The size in bytes of the table block when decompressed.
UInt64	8 bytes	cmpCrc64	The CRC64-ECMA checksum of the compressed table that follows this header.
UInt64	8 bytes	crc64	The CRC64-ECMA checksum of the decompressed table.

Each entry in the table points to a data block and to a specific item contained inside it.

Entry number 0 of the deduplication table points to the data belonging to LBA 0 of the media, and so on.

The entry is calculated as *(byte offset of data block in file << shift) + item number in data block*. Therefore a pointer with a raw value of 0x8003 in a deduplication table with a shift value of 5 indicates the third item in the data block located at absolute byte offset 1024 in the file.

A special case are the deduplication tables with data types *CdSectorPrefixCorrected* and *CdSectorSuffixCorrected*. In their case a mask of 0x00FFFFFF is used to obtain the pointer and a mask of 0xFF000000 is used to obtain the following flags:

Flag	Value	Description
None	0x00000000	The suffix or prefix cannot be regenerated as is stored in the pointed data block.
NotDumped	0x10000000	The sector has not been dumped. Ignore the pointer.
Correct	0x20000000	The suffix (only for MODE 1 sectors) or prefix is correct and can be regenerated. Ignore the pointer.
Mode2Form1Ok	0x30000000	The suffix for MODE 2 sectors is correct, can be regenerated, and corresponds to a MODE 2 Form 1 sector.
Mode2Form2Ok	0x40000000	The suffix for MODE 2 sectors is correct, can be regenerated, and corresponds to a MODE 2 Form 2 sector with a valid CRC.
Mode2Form2NoCrc	0x50000000	The suffix for MODE 2 sectors is correct, can be regenerated, and corresponds to a MODE 2 Form 2 sector with an empty CRC.

The geometry block ('GEOM')

The geometry block stores the geometry information of the media to allow for CHS<->LBA transformations.

This information is not necessarily physically correct, but may be whatever translation the drive had when it was dumped.

Type	Size	Name	Description
UInt32	4 bytes	identifier	The deduplication table identifier, always 'GEOM'
UInt32	4 bytes	cylinders	The number of cylinders.
UInt32	4 bytes	heads	The number of heads.
UInt32	4 bytes	sectorsPerTrack	The number of sectors per track.

The metadata block ('META')

The metadata block stores information about the media, that is not data in the media itself, like manufacturer, model, sequence, etc. All strings in this block are stored as little-endian Unicode's UTF-16 null-terminated.

Type	Size	Name	Description
UInt32	4 bytes	identifier	The deduplication table identifier, always 'META'
UInt32	4 bytes	blockSize	The size of this block including all of its data.
SInt32	4 bytes	mediaSequence	The number of heads.
SInt32	4 bytes	lastMediaSequence	The number of sectors per track.
UInt32	4 bytes	creatorOffset	Offset to start of creator string from start of this block.
UInt32	4 bytes	creatorLength	Length in bytes of the creator string.
UInt32	4 bytes	commentsOffset	Offset to start of comments string from start of this block.
UInt32	4 bytes	commentsLength	Length in bytes of the comments string.
UInt32	4 bytes	mediaTitleOffset	Offset to start of media title string from start of this block.
UInt32	4 bytes	mediaTitleLength	Length in bytes of the media title string.
UInt32	4 bytes	mediaManufacturerOffset	Offset to start of media manufacturer string from start of this block.
UInt32	4 bytes	mediaManufacturerLength	Length in bytes of the media manufacturer string.
UInt32	4 bytes	mediaModelOffset	Offset to start of media model string from start of this block.
UInt32	4 bytes	mediaModelLength	Length in bytes of the media model string.
UInt32	4 bytes	mediaSerialNumberOffset	Offset to start of media serial number string from start of this block.
UInt32	4 bytes	mediaSerialNumberLength	Length in bytes of the media serial number string.
UInt32	4 bytes	mediaBarcodeOffset	Offset to start of media barcode string from start of this block.
UInt32	4 bytes	mediaBarcodeLength	Length in bytes of the media barcode string.
UInt32	4 bytes	mediaPartNumberOffset	Offset to start of media part number string from start of this block.
UInt32	4 bytes	mediaPartNumberLength	Length in bytes of the media part number string.
UInt32	4 bytes	driveManufacturerOffset	Offset to start of drive manufacturer string from start of this block.
UInt32	4 bytes	driveManufacturerLength	Length in bytes of the drive manufacturer string.
UInt32	4 bytes	driveModelOffset	Offset to start of drive model string from start of this block.
UInt32	4 bytes	driveModelLength	Length in bytes of the drive model string.
UInt32	4 bytes	driveSerialNumberOffset	Offset to start of drive serial number string from start of this block.
UInt32	4 bytes	driveSerialNumberLength	Length in bytes of the drive serial number string.
UInt32	4 bytes	driveFirmwareRevisionOffset	Offset to start of drive firmware revision string from start of this block.
UInt32	4 bytes	driveFirmwareRevisionLength	Length in bytes of the drive firmware revision string.

The tracks block ('TRKS')

The tracks block contains a list of tracks as defined in the table of contents, or equivalent structure, usually found in optical discs (like CD, DVD, etc).

Type	Size	Name	Description
UInt32	4 bytes	identifier	The tracks block identifier, always 'TRKS'
UInt16	2 bytes	entries	The number of entries following this header
UInt64	8 bytes	crc64	CRC64-ECMA checksum of the entries following this header

Track entries

Type	Size	Name	Description
UInt8	1 byte	sequence	Track number.
UInt8	1 byte	type	Track type (see table below).
SInt64	8 bytes	start	Track starting LBA (including pregap).
SInt64	8 bytes	end	Track ending LBA.
SInt64	8 bytes	pregap	Size of track's pregap in sectors.
UInt8	1 byte	session	Session the track belongs to.
StringA	13 bytes	isrc	Track's ISRC in ASCIIZ.
UInt8	1 byte	flags	Track flags as indicated in TOC if applicable.

Track types

Type	Value	Description
Audio	0	All sectors in the track contain audio as defined by the Red Book.
Data	1	All sectors in the track contain user data that is not defined by any of the following types.
CdMode1	2	All sectors in the track contain user data according to MODE 1 as defined by the Yellow Book.
CdMode2Formless	3	All sectors in the track contain user data according to MODE 2 as defined by the Yellow and Green Books. Not all sectors belong to the same Form.
CdMode2Form1	4	All sectors in the track contain user data according to MODE 2 Form 1 as defined by the Yellow and Green Books. All sectors belong to the same Form.
CdMode2Form2	5	All sectors in the track contain user data according to MODE 2 Form 2 as defined by the Yellow and Green Books. All sectors belong to the same Form.

The CICM XML metadata block ('CICM')

This header indicates the presence of an embedded CICM XML metadata sidecar. It is stored as is without taking into consideration its contents.

Type	Size	Name	Description
UInt32	4 bytes	identifier	The deduplication table identifier, always 'CICM'
UInt32	4 bytes	length	The size in bytes of the embedded CICM XML metadata that follows this header.

The checksum block ('CKSM')

This block contains an array of checksums of the user data contained in this image. In the case of CompactDisc and similar media the checksums are applied to the full, including prefix and sufix, sector (2352 bytes).

When an image is altered this block must be considered stale and removed or omitted from the latest index as appropriate.

Type	Size	Name	Description
UInt32	4 bytes	identifier	The tracks block identifier, always 'CKSM'
UInt32	4 bytes	length	The length in bytes of the data following this header.
UInt16	2 bytes	entries	The number of entries following this header

Checksum entries

Type	Size	Name	Description
UInt8	1 byte	type	Checksum algorithm.
UInt32	4 bytes	length	Size in bytes of the checksum that immediately follows this entry.

Checksum entries

Name	Value	Description
Invalid	0	Invalid checksum entry, skip.
Md5	1	MD5
Sha1	2	SHA1
Sha256	3	SHA-256
SpamSum	4	SpamSum

The data position measurement block ('DPM')*

The data position measurement block stores information about the physical structure of a disc my taking measurements on the position of each sector.

The format of this block is pending to be defined in a future version of this specification.

The snapshot block ('SNAP')

The snapshot block contains a list of old indexes that are used as snapshots in time of old versions of the media contained by this image.

The format of this block is pending to be defined in a future version of this specification.

The parent file block ('PRNT')

This block contains information needed to find the parent image file from which this one is derived. All sectors that are found as non-written in this image should be read on the parent image.

The format of this block is pending to be defined in a future version of this specification.

The dump hardware block ('DMP')*

This block contains an array of hardware used to dump the media as well as a list of extents dumped by each one.

Type	Size	Name	Description
UInt32	4 bytes	identifier	The tracks block identifier, always 'DMP*'
UInt16	2 bytes	entries	The number of entries following this header
UInt32	4 bytes	length	The length in bytes of the data following this header.
UInt64	8 bytes	crc64	The CRC64-ECMA checksum of the data following this header

Dump hardware entries

Type	Size	Name	Description
UInt32	4 bytes	manufacturerLength	Checksum algorithm.
UInt32	4 bytes	modelLength	Length of UTF-8 manufacturer string.
UInt32	4 bytes	revisionLength	Length of UTF-8 revision string.
UInt32	4 bytes	firmwareLength	Length of UTF-8 firmware version string.
UInt32	4 bytes	serialLength	Length of UTF-8 serial number string.
UInt32	4 bytes	softwareNameLength	Length of UTF-8 software name string.
UInt32	4 bytes	softwareVersionLength	Length of UTF-8 software version string.
UInt32	4 bytes	softwareOperatingSystemLength	Length of UTF-8 software operating system string.
UInt32	4 bytes	extents	How many extents are after the strings.

Dump hardware extents

Type	Size	Name	Description
UInt64	8 bytes	start	Starting LBA of the extent (inclusive).
UInt64	8 bytes	end	Ending LBA of the extent (inclusive).

Each dump hardware entry is followed by the strings in the following order: manufacturer, model, revision, firmware version, serial number, software name, software version, software operating system.

The exents follow the last written string.

The tape file block ('TFLE')

This block lists all tape files. Tape files are separations written to some media, usually sequentially written digital tapes, that are called “filemarks”.

Type	Size	Name	Description
UInt32	4 bytes	identifier	The tracks block identifier, always 'TFLE'
UInt16	2 bytes	entries	The number of entries following this header
UInt32	4 bytes	length	The length in bytes of the data following this header.
UInt64	8 bytes	crc64	The CRC64-ECMA checksum of the data following this header

Tape file entries

Type	Size	Name	Description
UInt32	4 bytes	file	File number.
UInt8	1 byte	partition	Partition number this file belongs to.
UInt64	8 bytes	firstBlock	First block number, inclusive, of the file.
UInt64	8 bytes	lastBlock	Last block number, inclusive, of the file.

The tape partition block ('TPBT')

This block lists all tape files. Tape partitions are separations written to some media. They're used to separate two sets of data related but distant enough to be in the same tape. A famous example of using them is the LTFS filesystem.

Type	Size	Name	Description
UInt32	4 bytes	identifier	The tracks block identifier, always 'TPBT'
UInt16	2 bytes	entries	The number of entries following this header
UInt32	4 bytes	length	The length in bytes of the data following this header.
UInt64	8 bytes	crc64	The CRC64-ECMA checksum of the data following this header

Tape file entries

Type	Size	Name	Description
UInt8	1 byte	number	Partition number.
UInt64	8 bytes	firstBlock	First block number, inclusive, of the partition.
UInt64	8 bytes	lastBlock	Last block number, inclusive, of the partition.

The compact disc indexes block ('CDIX')

On CompactDisc and derived media, a track can have several indexes. They are used in some discs as a way of marking separations in the data, like different parts of a musical performance.

The table of contents always points to index 1, and all other indexes, including 0 (the pregap), are found on the subchannel information.

This block contains a list of all known indexes for fast lookup.

Type	Size	Name	Description
UInt32	4 bytes	identifier	The compact disc indexes block identifier, always 'CDIX'
UInt16	2 bytes	entries	The number of entries following this header
UInt32	4 bytes	length	The length in bytes of the data following this header.
UInt64	8 bytes	crc64	The CRC64-ECMA checksum of the data following this header

Compact disc index entries

Type	Size	Name	Description
UInt16	2 bytes	track	Track this index belongs to.
UInt16	2 bytes	index	Index number.
SInt32	4 bytes	lba	LBA where this index starts.

Annex A. Media types

This is a list of all known media types as of writing this specification. This list is not to be considered complete, being the source of *libaaruformat* the origin of the most up-to-date list.

Enum	Value	Summary
Unknown	0	Unknown disk type
UnknownMO	1	Unknown magneto-optical
GENERIC_HDD	2	Generic hard disk
Microdrive	3	Microdrive type hard disk
Zone_HDD	4	Zoned hard disk
FlashDrive	5	USB flash drives
UnknownTape	6	Unknown data tape
CD	10	Any unknown or standard violating CD
CDDA	11	CD Digital Audio (Red Book)
CDG	12	CD+G (Red Book)
CDEG	13	CD+EG (Red Book)
CDI	14	CD-i (Green Book)
CDROM	15	CD-ROM (Yellow Book)
CDROMXA	16	CD-ROM XA (Yellow Book)
CDPLUS	17	CD+ (Blue Book)
CDMO	18	CD-MO (Orange Book)
CDR	19	CD-Recordable (Orange Book)
CDRW	20	CD-ReWritable (Orange Book)
CDMRW	21	Mount-Rainier CD-RW
VCD	22	Video CD (White Book)
SVCD	23	Super Video CD (White Book)
PCD	24	Photo CD (Beige Book)
SACD	25	Super Audio CD (Scarlet Book)
DDCD	26	Double-Density CD-ROM (Purple Book)
DDCDR	27	DD CD-R (Purple Book)
DDCDRW	28	DD CD-RW (Purple Book)
DTSCD	29	DTS audio CD (non-standard)
CDMIDI	30	CD-MIDI (Red Book)
CDV	31	CD-Video (ISO/IEC 61104)
PD650	32	120mm, Phase-Change, 1298496 sectors, 512 bytes/sector, PD650, ECMA-240, ISO 15485
PD650_WORM	33	120mm, Write-Once, 1281856 sectors, 512 bytes/sector, PD650, ECMA-240, ISO 15485
CDIREADY	34	CD-i Ready, contains a track before the first TOC track, in mode 2, and all TOC tracks are Audio. Subchannel marks track as audio pause.
FMTOWNS	35	
DVDROM	40	DVD-ROM (applies to DVD Video and DVD Audio)
DVDR	41	DVD-R
DVDRW	42	DVD-RW
DVDPR	43	DVD+R
DVDPRW	44	DVD+RW
DVDPRWDL	45	DVD+RW DL
DVDRDL	46	DVD-R DL
DVDPRDL	47	DVD+R DL
DVDRAM	48	DVD-RAM

Enum	Value	Summary
DVDRWDL	49	DVD-RW DL
DVDDownload	50	DVD-Download
HDDVDROM	51	HD DVD-ROM (applies to HD DVD Video)
HDDVDRAM	52	HD DVD-RAM
HDDVDR	53	HD DVD-R
HDDVDRW	54	HD DVD-RW
HDDVDRDL	55	HD DVD-R DL
HDDVDRWDL	56	HD DVD-RW DL
BDROM	60	BD-ROM (and BD Video)
BDR	61	BD-R
BDRE	62	BD-RE
BDRXL	63	BD-R XL
BDREXL	64	BD-RE XL
UHDBD	65	Ultra HD Blu-ray
EVD	70	Enhanced Versatile Disc
FVD	71	Forward Versatile Disc
HVD	72	Holographic Versatile Disc
CBHD	73	China Blue High Definition
HDVMD	74	High Definition Versatile Multilayer Disc
VCDHD	75	Versatile Compact Disc High Density
SVOD	76	Stacked Volumetric Optical Disc
FDDVD	77	Five Dimensional disc
CVD	78	China Video Disc
LD	80	Pioneer LaserDisc
LDROM	81	Pioneer LaserDisc data
LDROM2	82	
LVROM	83	
MegaLD	84	
CRVdisc	85	Writable LaserDisc with support for component video
HiMD	90	Sony Hi-MD
MD	91	Sony MiniDisc
MDData	92	Sony MD-Data
MDData2	93	Sony MD-Data2
MD60	94	Sony MiniDisc, 60 minutes, formatted with Hi-MD format
MD74	95	Sony MiniDisc, 74 minutes, formatted with Hi-MD format
MD80	96	Sony MiniDisc, 80 minutes, formatted with Hi-MD format
UDO	100	5.25", Phase-Change, 1834348 sectors, 8192 bytes/sector, Ultra Density Optical, ECMA-350, ISO 17345
UDO2	101	5.25", Phase-Change, 3669724 sectors, 8192 bytes/sector, Ultra Density Optical 2, ECMA-380, ISO 11976
UDO2_WORM	102	5.25", Write-Once, 3668759 sectors, 8192 bytes/sector, Ultra Density Optical 2, ECMA-380, ISO 11976
PlayStationMemoryCard	110	
PlayStationMemoryCard2	111	
PS1CD	112	Sony PlayStation game CD
PS2CD	113	Sony PlayStation 2 game CD
PS2DVD	114	Sony PlayStation 2 game DVD
PS3DVD	115	Sony PlayStation 3 game DVD
PS3BD	116	Sony PlayStation 3 game Blu-ray
PS4BD	117	Sony PlayStation 4 game Blu-ray
UMD	118	Sony PlayStation Portable Universal Media Disc (ECMA-365)
PlayStationVitaGameCard	119	
PS5BD	120	Sony PlayStation 5 game Ultra HD Blu-ray
XGD	130	Microsoft X-box Game Disc

Enum	Value	Summary
XGD2	131	Microsoft X-box 360 Game Disc
XGD3	132	Microsoft X-box 360 Game Disc
XGD4	133	Microsoft X-box One Game Disc
MEGACD	150	Sega MegaCD
SATURNCD	151	Sega Saturn disc
GDROM	152	Sega/Yamaha Gigabyte Disc
GDR	153	Sega/Yamaha recordable Gigabyte Disc
SegaCard	154	
MilCD	155	
MegaDriveCartridge	156	
_32XCartridge	157	
SegaPicoCartridge	158	
MasterSystemCartridge	159	
GameGearCartridge	160	
SegaSaturnCartridge	161	
HuCard	170	PC-Engine / TurboGrafx cartridge
SuperCDROM2	171	PC-Engine / TurboGrafx CD
JaguarCD	172	Atari Jaguar CD
ThreeDO	173	3DO CD
PCFX	174	NEC PC-FX
NeoGeoCD	175	NEO-GEO CD
CDTV	176	Commodore CDTV
CD32	177	Amiga CD32
Nuon	178	Nuon (DVD based videogame console)
Playdia	179	Bandai Playdia
Apple32SS	180	5.25", SS, DD, 35 tracks, 13 spt, 256 bytes/sector, GCR
Apple32DS	181	5.25", DS, DD, 35 tracks, 13 spt, 256 bytes/sector, GCR
Apple33SS	182	5.25", SS, DD, 35 tracks, 16 spt, 256 bytes/sector, GCR
Apple33DS	183	5.25", DS, DD, 35 tracks, 16 spt, 256 bytes/sector, GCR
AppleSonySS	184	3.5", SS, DD, 80 tracks, 8 to 12 spt, 512 bytes/sector, GCR
AppleSonyDS	185	3.5", DS, DD, 80 tracks, 8 to 12 spt, 512 bytes/sector, GCR
AppleFileWare	186	5.25", DS, ?D, ?? tracks, ?? spt, 512 bytes/sector, GCR, opposite side heads, aka Twiggy
DOS_525_SS_DD_8	190	5.25", SS, DD, 40 tracks, 8 spt, 512 bytes/sector, MFM
DOS_525_SS_DD_9	191	5.25", SS, DD, 40 tracks, 9 spt, 512 bytes/sector, MFM
DOS_525_DS_DD_8	192	5.25", DS, DD, 40 tracks, 8 spt, 512 bytes/sector, MFM
DOS_525_DS_DD_9	193	5.25", DS, DD, 40 tracks, 9 spt, 512 bytes/sector, MFM
DOS_525_HD	194	5.25", DS, HD, 80 tracks, 15 spt, 512 bytes/sector, MFM
DOS_35_SS_DD_8	195	3.5", SS, DD, 80 tracks, 8 spt, 512 bytes/sector, MFM
DOS_35_SS_DD_9	196	3.5", SS, DD, 80 tracks, 9 spt, 512 bytes/sector, MFM
DOS_35_DS_DD_8	197	3.5", DS, DD, 80 tracks, 8 spt, 512 bytes/sector, MFM
DOS_35_DS_DD_9	198	3.5", DS, DD, 80 tracks, 9 spt, 512 bytes/sector, MFM
DOS_35_HD	199	3.5", DS, HD, 80 tracks, 18 spt, 512 bytes/sector, MFM
DOS_35_ED	200	3.5", DS, ED, 80 tracks, 36 spt, 512 bytes/sector, MFM
DMF	201	3.5", DS, HD, 80 tracks, 21 spt, 512 bytes/sector, MFM
DMF_82	202	3.5", DS, HD, 82 tracks, 21 spt, 512 bytes/sector, MFM
XDF_525	203	5.25", DS, HD, 80 tracks, ? spt, ??? + ??? + ??? bytes/sector, MFM track 0 = ??15 sectors, 512 bytes/sector, falsified to DOS as 19 spt, 512 bps
XDF_35	204	3.5", DS, HD, 80 tracks, 4 spt, 8192 + 2048 + 1024 + 512 bytes/sector, MFM track 0 = 19 sectors, 512 bytes/sector, falsified to DOS as 23 spt, 512 bps
IBM23FD	210	8", SS, SD, 32 tracks, 8 spt, 319 bytes/sector, FM
IBM33FD_128	211	8", SS, SD, 73 tracks, 26 spt, 128 bytes/sector, FM
IBM33FD_256	212	8", SS, SD, 74 tracks, 15 spt, 256 bytes/sector, FM, track 0 = 26 sectors, 128 bytes/sector

Enum	Value	Summary
IBM33FD_512	213	8", SS, SD, 74 tracks, 8 spt, 512 bytes/sector, FM, track 0 = 26 sectors, 128 bytes/sector
IBM43FD_128	214	8", DS, SD, 74 tracks, 26 spt, 128 bytes/sector, FM, track 0 = 26 sectors, 128 bytes/sector
IBM43FD_256	215	8", DS, SD, 74 tracks, 26 spt, 256 bytes/sector, FM, track 0 = 26 sectors, 128 bytes/sector
IBM53FD_256	216	8", DS, DD, 74 tracks, 26 spt, 256 bytes/sector, MFM, track 0 side 0 = 26 sectors, 128 bytes/sector, track 0 side 1 = 26 sectors, 256 bytes/sector
IBM53FD_512	217	8", DS, DD, 74 tracks, 15 spt, 512 bytes/sector, MFM, track 0 side 0 = 26 sectors, 128 bytes/sector, track 0 side 1 = 26 sectors, 256 bytes/sector
IBM53FD_1024	218	8", DS, DD, 74 tracks, 8 spt, 1024 bytes/sector, MFM, track 0 side 0 = 26 sectors, 128 bytes/sector, track 0 side 1 = 26 sectors, 256 bytes/sector
RX01	220	8", SS, DD, 77 tracks, 26 spt, 128 bytes/sector, FM
RX02	221	8", SS, DD, 77 tracks, 26 spt, 256 bytes/sector, FM/MFM
RX03	222	8", DS, DD, 77 tracks, 26 spt, 256 bytes/sector, FM/MFM
RX50	223	5.25", SS, DD, 80 tracks, 10 spt, 512 bytes/sector, MFM
ACORN_525_SS_SD_40	230	5,25", SS, SD, 40 tracks, 10 spt, 256 bytes/sector, FM
ACORN_525_SS_SD_80	231	5,25", SS, SD, 80 tracks, 10 spt, 256 bytes/sector, FM
ACORN_525_SS_DD_40	232	5,25", SS, DD, 40 tracks, 16 spt, 256 bytes/sector, MFM
ACORN_525_SS_DD_80	233	5,25", SS, DD, 80 tracks, 16 spt, 256 bytes/sector, MFM
ACORN_525_DS_DD	234	5,25", DS, DD, 80 tracks, 16 spt, 256 bytes/sector, MFM
ACORN_35_DS_DD	235	3,5", DS, DD, 80 tracks, 5 spt, 1024 bytes/sector, MFM
ACORN_35_DS_HD	236	3,5", DS, HD, 80 tracks, 10 spt, 1024 bytes/sector, MFM
ATARI_525_SD	240	5,25", SS, SD, 40 tracks, 18 spt, 128 bytes/sector, FM
ATARI_525_ED	241	5,25", SS, ED, 40 tracks, 26 spt, 128 bytes/sector, MFM
ATARI_525_DD	242	5,25", SS, DD, 40 tracks, 18 spt, 256 bytes/sector, MFM
ATARI_35_SS_DD	243	3,5", SS, DD, 80 tracks, 10 spt, 512 bytes/sector, MFM
ATARI_35_DS_DD	244	3,5", DS, DD, 80 tracks, 10 spt, 512 bytes/sector, MFM
ATARI_35_SS_DD_11	245	3,5", SS, DD, 80 tracks, 11 spt, 512 bytes/sector, MFM
ATARI_35_DS_DD_11	246	3,5", DS, DD, 80 tracks, 11 spt, 512 bytes/sector, MFM
CBM_35_DD	250	3,5", DS, DD, 80 tracks, 10 spt, 512 bytes/sector, MFM (1581)
CBM_AMIGA_35_DD	251	3,5", DS, DD, 80 tracks, 11 spt, 512 bytes/sector, MFM (Amiga)
CBM_AMIGA_35_HD	252	3,5", DS, HD, 80 tracks, 22 spt, 512 bytes/sector, MFM (Amiga)
CBM_1540	253	5,25", SS, DD, 35 tracks, GCR
CBM_1540_Ext	254	5,25", SS, DD, 40 tracks, GCR
CBM_1571	255	5,25", DS, DD, 35 tracks, GCR
NEC_8_SD	260	8", DS, SD, 77 tracks, 26 spt, 128 bytes/sector, FM
NEC_8_DD	261	8", DS, DD, 77 tracks, 26 spt, 256 bytes/sector, MFM
NEC_525_SS	262	5.25", SS, SD, 80 tracks, 16 spt, 256 bytes/sector, FM
NEC_525_DS	263	5.25", DS, SD, 80 tracks, 16 spt, 256 bytes/sector, MFM
NEC_525_HD	264	5,25", DS, HD, 77 tracks, 8 spt, 1024 bytes/sector, MFM
NEC_35_HD_8	265	3,5", DS, HD, 77 tracks, 8 spt, 1024 bytes/sector, MFM, aka mode 3
NEC_35_HD_15	266	3,5", DS, HD, 80 tracks, 15 spt, 512 bytes/sector, MFM
NEC_35_TD	267	3,5", DS, TD, 240 tracks, 38 spt, 512 bytes/sector, MFM
SHARP_525	264	5,25", DS, HD, 77 tracks, 8 spt, 1024 bytes/sector, MFM
SHARP_525_9	268	3,5", DS, HD, 80 tracks, 9 spt, 1024 bytes/sector, MFM
SHARP_35	265	3,5", DS, HD, 77 tracks, 8 spt, 1024 bytes/sector, MFM
SHARP_35_9	269	3,5", DS, HD, 80 tracks, 9 spt, 1024 bytes/sector, MFM
ECMA_99_8	270	5,25", DS, DD, 80 tracks, 8 spt, 1024 bytes/sector, MFM, track 0 side 0 = 26 sectors, 128 bytes/sector, track 0 side 1 = 26 sectors, 256 bytes/sector
ECMA_99_15	271	5,25", DS, DD, 77 tracks, 15 spt, 512 bytes/sector, MFM, track 0 side 0 = 26 sectors, 128 bytes/sector, track 0 side 1 = 26 sectors, 256 bytes/sector
ECMA_99_26	272	5,25", DS, DD, 77 tracks, 26 spt, 256 bytes/sector, MFM, track 0 side 0 = 26 sectors, 128 bytes/sector, track 0 side 1 = 26 sectors, 256 bytes/sector

Enum	Value	Summary
ECMA_100	198	3,5", DS, DD, 80 tracks, 9 spt, 512 bytes/sector, MFM
ECMA_125	199	3,5", DS, HD, 80 tracks, 18 spt, 512 bytes/sector, MFM
ECMA_147	200	3,5", DS, ED, 80 tracks, 36 spt, 512 bytes/sector, MFM
ECMA_54	273	8", SS, SD, 77 tracks, 26 spt, 128 bytes/sector, FM
ECMA_59	274	8", DS, SD, 77 tracks, 26 spt, 128 bytes/sector, FM
ECMA_66	275	5,25", SS, DD, 35 tracks, 9 spt, 256 bytes/sector, FM, track 0 side 0 = 16 sectors, 128 bytes/sector
ECMA_69_8	276	8", DS, DD, 77 tracks, 8 spt, 1024 bytes/sector, FM, track 0 side 0 = 26 sectors, 128 bytes/sector, track 0 side 1 = 26 sectors, 256 bytes/sector
ECMA_69_15	277	8", DS, DD, 77 tracks, 15 spt, 512 bytes/sector, FM, track 0 side 0 = 26 sectors, 128 bytes/sector, track 0 side 1 = 26 sectors, 256 bytes/sector
ECMA_69_26	278	8", DS, DD, 77 tracks, 26 spt, 256 bytes/sector, FM, track 0 side 0 = 26 sectors, 128 bytes/sector, track 0 side 1 = 26 sectors, 256 bytes/sector
ECMA_70	279	5,25", DS, DD, 40 tracks, 16 spt, 256 bytes/sector, FM, track 0 side 0 = 16 sectors, 128 bytes/sector, track 0 side 1 = 16 sectors, 256 bytes/sector
ECMA_78	280	5,25", DS, DD, 80 tracks, 16 spt, 256 bytes/sector, FM, track 0 side 0 = 16 sectors, 128 bytes/sector, track 0 side 1 = 16 sectors, 256 bytes/sector
ECMA_78_2	281	5,25", DS, DD, 80 tracks, 9 spt, 512 bytes/sector, FM
FDFORMAT_525_DD	290	5,25", DS, DD, 82 tracks, 10 spt, 512 bytes/sector, MFM
FDFORMAT_525_HD	291	5,25", DS, HD, 82 tracks, 17 spt, 512 bytes/sector, MFM
FDFORMAT_35_DD	292	3,5", DS, DD, 82 tracks, 10 spt, 512 bytes/sector, MFM
FDFORMAT_35_HD	293	3,5", DS, HD, 82 tracks, 21 spt, 512 bytes/sector, MFM
Apricot_35	309	3.5", DS, DD, 70 tracks, 9 spt, 512 bytes/sector, MFM
ADR2120	310	
ADR260	311	
ADR30	312	
ADR50	313	
AIT1	320	
AIT1Turbo	321	
AIT2	322	
AIT2Turbo	323	
AIT3	324	
AIT3Ex	325	
AIT3Turbo	326	
AIT4	327	
AIT5	328	
AITETurbo	329	
SAIT1	330	
SAIT2	331	
Bernoulli	340	Obsolete type for 8"x11" Bernoulli Box disk
Bernoulli2	341	Obsolete type for 5⅓" Bernoulli Box II disks
Ditto	342	
DittoMax	343	
Jaz	344	
Jaz2	345	
PocketZip	346	
REV120	347	
REV35	348	
REV70	349	
ZIP100	350	
ZIP250	351	
ZIP750	352	
Bernoulli35	353	5⅓" Bernoulli Box II disk with 35Mb capacity
Bernoulli44	354	5⅓" Bernoulli Box II disk with 44Mb capacity

Enum	Value	Summary
Bernoulli65	355	5⅓" Bernoulli Box II disk with 65Mb capacity
Bernoulli90	356	5⅓" Bernoulli Box II disk with 90Mb capacity
Bernoulli105	357	5⅓" Bernoulli Box II disk with 105Mb capacity
Bernoulli150	358	5⅓" Bernoulli Box II disk with 150Mb capacity
Bernoulli230	359	5⅓" Bernoulli Box II disk with 230Mb capacity
CompactCassette	360	
Data8	361	
MiniDV	362	
Dcas25	363	D/CAS-25: Digital data on Compact Cassette form factor, special magnetic media, 9-track
Dcas85	364	D/CAS-85: Digital data on Compact Cassette form factor, special magnetic media, 17-track
Dcas103	365	D/CAS-103: Digital data on Compact Cassette form factor, special magnetic media, 21-track
CFast	370	
CompactFlash	371	
CompactFlashType2	372	
DigitalAudioTape	380	
DAT160	381	
DAT320	382	
DAT72	383	
DDS1	384	
DDS2	385	
DDS3	386	
DDS4	387	
CompactTapeI	390	
CompactTapeII	391	
DECtapeII	392	
DLTtapeIII	393	
DLTtapeIIIxt	394	
DLTtapeIV	395	
DLTtapeS4	396	
SDLT1	397	
SDLT2	398	
VStapeI	399	
Exatape15m	400	
Exatape22m	401	
Exatape22mAME	402	
Exatape28m	403	
Exatape40m	404	
Exatape45m	405	
Exatape54m	406	
Exatape75m	407	
Exatape76m	408	
Exatape80m	409	
Exatape106m	410	
Exatape160mXL	411	
Exatape112m	412	
Exatape125m	413	
Exatape150m	414	
Exatape170m	415	
Exatape225m	416	
ExpressCard34	420	
ExpressCard54	421	

Enum	Value	Summary
PCCardTypeI	422	
PCCardTypeII	423	
PCCardTypeIII	424	
PCCardTypeIV	425	
EZ135	430	SyQuest 135Mb cartridge for use in EZ135 and EZFlyer drives
EZ230	431	SyQuest EZFlyer 230Mb cartridge for use in EZFlyer drive
Quest	432	SyQuest 4.7Gb for use in Quest drive
SparQ	433	SyQuest SparQ 1Gb cartridge
SQ100	434	SyQuest 5Mb cartridge for SQ306RD drive
SQ200	435	SyQuest 10Mb cartridge for SQ312RD drive
SQ300	436	SyQuest 15Mb cartridge for SQ319RD drive
SQ310	437	SyQuest 105Mb cartridge for SQ3105 and SQ3270 drives
SQ327	438	SyQuest 270Mb cartridge for SQ3270 drive
SQ400	439	SyQuest 44Mb cartridge for SQ555, SQ5110 and SQ5200C/SQ200 drives
SQ800	440	SyQuest 88Mb cartridge for SQ5110 and SQ5200C/SQ200 drives
SQ1500	441	SyQuest 1.5Gb cartridge for SyJet drive
SQ2000	442	SyQuest 200Mb cartridge for use in SQ5200C drive
SyJet	443	SyQuest 1.5Gb cartridge for SyJet drive
FamicomGamePak	450	
GameBoyAdvanceGamePak	451	
GameBoyGamePak	452	
GOD	453	Nintendo GameCube Optical Disc
N64DD	454	
N64GamePak	455	
NESGamePak	456	
Nintendo3DSGameCard	457	
NintendoDiskCard	458	
NintendoDSGameCard	459	
NintendoDSiGameCard	460	
SNESGamePak	461	
SNESGamePakUS	462	
WOD	463	Nintendo Wii Optical Disc
WUOD	464	Nintendo Wii U Optical Disc
SwitchGameCard	465	
IBM3470	470	
IBM3480	471	
IBM3490	472	
IBM3490E	473	
IBM3592	474	
LTO	480	
LTO2	481	
LTO3	482	
LTO3WORM	483	
LTO4	484	
LTO4WORM	485	
LTO5	486	
LTO5WORM	487	
LTO6	488	
LTO6WORM	489	
LTO7	490	
LTO7WORM	491	
MemoryStick	510	
MemoryStickDuo	511	
MemoryStickMicro	512	

Enum	Value	Summary
MemoryStickPro	513	
MemoryStickProDuo	514	
microSD	520	
miniSD	521	
SecureDigital	522	
MMC	530	
MMCmicro	531	
RSMMC	532	
MMCplus	533	
MMCmobile	534	
MLR1	540	
MLR1SL	541	
MLR3	542	
SLR1	543	
SLR2	544	
SLR3	545	
SLR32	546	
SLR32SL	547	
SLR4	548	
SLR5	549	
SLR5SL	550	
SLR6	551	
SLRtape7	552	
SLRtape7SL	553	
SLRtape24	554	
SLRtape24SL	555	
SLRtape40	556	
SLRtape50	557	
SLRtape60	558	
SLRtape75	559	
SLRtape100	560	
SLRtape140	561	
QIC11	570	
QIC120	571	
QIC1350	572	
QIC150	573	
QIC24	574	
QIC3010	575	
QIC3020	576	
QIC3080	577	
QIC3095	578	
QIC320	579	
QIC40	580	
QIC525	581	
QIC80	582	
STK4480	590	
STK4490	591	
STK9490	592	
T9840A	593	
T9840B	594	
T9840C	595	
T9840D	596	
T9940A	597	
T9940B	598	

Enum	Value	Summary
T10000A	599	
T10000B	600	
T10000C	601	
T10000D	602	
Travan	610	
Travan1Ex	611	
Travan3	612	
Travan3Ex	613	
Travan4	614	
Travan5	615	
Travan7	616	
VXA1	620	
VXA2	621	
VXA3	622	
ECMA_153	630	5,25", M.O., WORM, 650Mb, 318750 sectors, 1024 bytes/sector, ECMA-153, ISO 11560
ECMA_153_512	631	5,25", M.O., WORM, 600Mb, 581250 sectors, 512 bytes/sector, ECMA-153, ISO 11560
ECMA_154	632	3,5", M.O., RW, 128Mb, 248826 sectors, 512 bytes/sector, ECMA-154, ISO 10090
ECMA_183_512	633	5,25", M.O., RW/WORM, 1Gb, 904995 sectors, 512 bytes/sector, ECMA-183, ISO 13481
ECMA_183	634	5,25", M.O., RW/WORM, 1Gb, 498526 sectors, 1024 bytes/sector, ECMA-183, ISO 13481
ECMA_184_512	635	5,25", M.O., RW/WORM, 1.2Gb, 1165600 sectors, 512 bytes/sector, ECMA-184, ISO 13549
ECMA_184	636	5,25", M.O., RW/WORM, 1.3Gb, 639200 sectors, 1024 bytes/sector, ECMA-184, ISO 13549
ECMA_189	637	300mm, M.O., WORM, ??? sectors, 1024 bytes/sector, ECMA-189, ISO 13614
ECMA_190	638	300mm, M.O., WORM, ??? sectors, 1024 bytes/sector, ECMA-190, ISO 13403
ECMA_195	639	5,25", M.O., RW/WORM, 936921 or 948770 sectors, 1024 bytes/sector, ECMA-195, ISO 13842
ECMA_195_512	640	5,25", M.O., RW/WORM, 1644581 or 1647371 sectors, 512 bytes/sector, ECMA-195, ISO 13842
ECMA_201	641	3,5", M.O., 446325 sectors, 512 bytes/sector, ECMA-201, ISO 13963
ECMA_201_ROM	642	3,5", M.O., 429975 sectors, 512 bytes/sector, embossed, ISO 13963
ECMA_223	643	3,5", M.O., 371371 sectors, 1024 bytes/sector, ECMA-223
ECMA_223_512	644	3,5", M.O., 694929 sectors, 512 bytes/sector, ECMA-223
ECMA_238	645	5,25", M.O., 1244621 sectors, 1024 bytes/sector, ECMA-238, ISO 15486
ECMA_239	646	3,5", M.O., 310352, 320332 or 321100 sectors, 2048 bytes/sector, ECMA-239, ISO 15498
ECMA_260	647	356mm, M.O., 14476734 sectors, 1024 bytes/sector, ECMA-260, ISO 15898
ECMA_260_Double	648	356mm, M.O., 24445990 sectors, 1024 bytes/sector, ECMA-260, ISO 15898
ECMA_280	649	5,25", M.O., 1128134 sectors, 2048 bytes/sector, ECMA-280, ISO 18093
ECMA_317	650	300mm, M.O., 7355716 sectors, 2048 bytes/sector, ECMA-317, ISO 20162
ECMA_322	651	5,25", M.O., 1095840 sectors, 4096 bytes/sector, ECMA-322, ISO 22092, 9.1Gb/cart
ECMA_322_2k	652	5,25", M.O., 2043664 sectors, 2048 bytes/sector, ECMA-322, ISO 22092, 8.6Gb/cart
GigaMo	653	3,5", M.O., 605846 sectors, 2048 bytes/sector, Cherry Book, GigaMo, ECMA-351, ISO 17346
GigaMo2	654	3,5", M.O., 1063146 sectors, 2048 bytes/sector, Cherry Book 2, GigaMo 2, ECMA-353, ISO 22533
ISO_15286	655	5,25", M.O., 1263472 sectors, 2048 bytes/sector, ISO 15286, 5.2Gb/cart

Enum	Value	Summary
ISO_15286_1024	656	5,25", M.O., 2319786 sectors, 1024 bytes/sector, ISO 15286, 4.8Gb/cart
ISO_15286_512	657	5,25", M.O., ??????? sectors, 512 bytes/sector, ISO 15286, 4.1Gb/cart
ISO_10089	658	5,25", M.O., 314569 sectors, 1024 bytes/sector, ISO 10089, 650Mb/cart
ISO_10089_512	659	5,25", M.O., ?????? sectors, 512 bytes/sector, ISO 10089, 594Mb/cart
CompactFloppy	660	
DemiDiskette	661	
Floptical	662	3.5", 652 tracks, 2 sides, 512 bytes/sector, Floptical, ECMA-207, ISO 14169
HiFD	663	
QuickDisk	664	
UHD144	665	
VideoFloppy	666	
Wafer	667	
ZXMicrodrive	668	
MetaFloppy_Mod_II	669	5.25", SS, DD, 77 tracks, 16 spt, 256 bytes/sector, MFM, 100 tpi, 300rpm
BeeCard	670	
Borsu	671	
DataStore	672	
DIR	673	
DST	674	
DTF	675	
DTF2	676	
Flextra3020	677	
Flextra3225	678	
HiTC1	679	
HiTC2	680	
LT1	681	
MiniCard	872	
Orb	683	
Orb5	684	
SmartMedia	685	
xD	686	
XQD	687	
DataPlay	688	
AppleProfile	690	
AppleWidget	691	
AppleHD20	692	
PriamDataTower	693	
Pippin	694	
RA60	700	2382 cylinders, 4 tracks/cylinder, 42 sectors/track, 128 words/sector, 32 bits/word, 512 bytes/sector, 204890112 bytes
RA80	701	546 cylinders, 14 tracks/cylinder, 31 sectors/track, 128 words/sector, 32 bits/word, 512 bytes/sector, 121325568 bytes
RA81	702	1248 cylinders, 14 tracks/cylinder, 51 sectors/track, 128 words/sector, 32 bits/word, 512 bytes/sector, 456228864 bytes
RC25	703	302 cylinders, 4 tracks/cylinder, 42 sectors/track, 128 words/sector, 32 bits/word, 512 bytes/sector, 25976832 bytes
RD31	704	615 cylinders, 4 tracks/cylinder, 17 sectors/track, 128 words/sector, 32 bits/word, 512 bytes/sector, 21411840 bytes
RD32	705	820 cylinders, 6 tracks/cylinder, 17 sectors/track, 128 words/sector, 32 bits/word, 512 bytes/sector, 42823680 bytes
RD51	706	306 cylinders, 4 tracks/cylinder, 17 sectors/track, 128 words/sector, 32 bits/word, 512 bytes/sector, 10653696 bytes
RD52	707	480 cylinders, 7 tracks/cylinder, 18 sectors/track, 128 words/sector, 32 bits/word, 512 bytes/sector, 30965760 bytes

Enum	Value	Summary
RD53	708	1024 cylinders, 7 tracks/cylinder, 18 sectors/track, 128 words/sector, 32 bits/word, 512 bytes/sector, 75497472 bytes
RD54	709	1225 cylinders, 8 tracks/cylinder, 18 sectors/track, 128 words/sector, 32 bits/word, 512 bytes/sector, 159936000 bytes
RK06	710	411 cylinders, 3 tracks/cylinder, 22 sectors/track, 256 words/sector, 16 bits/word, 512 bytes/sector, 13888512 bytes
RK06_18	711	411 cylinders, 3 tracks/cylinder, 20 sectors/track, 256 words/sector, 18 bits/word, 576 bytes/sector, 14204160 bytes
RK07	712	815 cylinders, 3 tracks/cylinder, 22 sectors/track, 256 words/sector, 16 bits/word, 512 bytes/sector, 27540480 bytes
RK07_18	713	815 cylinders, 3 tracks/cylinder, 20 sectors/track, 256 words/sector, 18 bits/word, 576 bytes/sector, 28166400 bytes
RM02	714	823 cylinders, 5 tracks/cylinder, 32 sectors/track, 128 words/sector, 32 bits/word, 512 bytes/sector, 67420160 bytes
RM03	715	823 cylinders, 5 tracks/cylinder, 32 sectors/track, 128 words/sector, 32 bits/word, 512 bytes/sector, 67420160 bytes
RM05	716	823 cylinders, 19 tracks/cylinder, 32 sectors/track, 128 words/sector, 32 bits/word, 512 bytes/sector, 256196608 bytes
RP02	717	203 cylinders, 10 tracks/cylinder, 22 sectors/track, 128 words/sector, 32 bits/word, 512 bytes/sector, 22865920 bytes
RP02_18	718	203 cylinders, 10 tracks/cylinder, 20 sectors/track, 128 words/sector, 36 bits/word, 576 bytes/sector, 23385600 bytes
RP03	719	400 cylinders, 10 tracks/cylinder, 22 sectors/track, 128 words/sector, 32 bits/word, 512 bytes/sector, 45056000 bytes
RP03_18	720	400 cylinders, 10 tracks/cylinder, 20 sectors/track, 128 words/sector, 36 bits/word, 576 bytes/sector, 46080000 bytes
RP04	721	411 cylinders, 19 tracks/cylinder, 22 sectors/track, 128 words/sector, 32 bits/word, 512 bytes/sector, 87960576 bytes
RP04_18	722	411 cylinders, 19 tracks/cylinder, 20 sectors/track, 128 words/sector, 36 bits/word, 576 bytes/sector, 89959680 bytes
RP05	723	411 cylinders, 19 tracks/cylinder, 22 sectors/track, 128 words/sector, 32 bits/word, 512 bytes/sector, 87960576 bytes
RP05_18	724	411 cylinders, 19 tracks/cylinder, 20 sectors/track, 128 words/sector, 36 bits/word, 576 bytes/sector, 89959680 bytes
RP06	725	815 cylinders, 19 tracks/cylinder, 22 sectors/track, 128 words/sector, 32 bits/word, 512 bytes/sector, 174423040 bytes
RP06_18	726	815 cylinders, 19 tracks/cylinder, 20 sectors/track, 128 words/sector, 36 bits/word, 576 bytes/sector, 178387200 bytes
LS120	730	
LS240	731	
FD32MB	732	
RDX	733	
RDX320	734	Imation 320Gb RDX
VideoNow	740	
VideoNowColor	741	
VideoNowXp	742	
Bernoulli10	750	8"x11" Bernoulli Box disk with 10Mb capacity
Bernoulli20	751	8"x11" Bernoulli Box disk with 20Mb capacity
BernoulliBox2_20	752	5⅓" Bernoulli Box II disk with 20Mb capacity
KodakVerbatim3	760	
KodakVerbatim6	761	
KodakVerbatim12	762	
ProfessionalDisc	770	Professional Disc for video, single layer, rewritable, 23Gb
ProfessionalDiscDual	771	Professional Disc for video, dual layer, rewritable, 50Gb

Enum	Value	Summary
ProfessionalDiscTriple	772	Professional Disc for video, triple layer, rewritable, 100Gb
ProfessionalDiscQuad	773	Professional Disc for video, quad layer, write once, 128Gb
PDD	774	Professional Disc for DATA, single layer, rewritable, 23Gb
PDD_WORM	775	Professional Disc for DATA, single layer, write once, 23Gb
ArchivalDisc	776	Archival Disc, 1st gen., 300Gb
ArchivalDisc2	777	Archival Disc, 2nd gen., 500Gb
ArchivalDisc3	778	Archival Disc, 3rd gen., 1Tb
ODC300R	779	Optical Disc archive, 1st gen., write once, 300Gb
ODC300RE	780	Optical Disc archive, 1st gen., rewritable, 300Gb
ODC600R	781	Optical Disc archive, 2nd gen., write once, 600Gb
ODC600RE	782	Optical Disc archive, 2nd gen., rewritable, 600Gb
ODC1200RE	783	Optical Disc archive, 3rd gen., rewritable, 1200Gb
ODC1500R	784	Optical Disc archive, 3rd gen., write once, 1500Gb
ODC3300R	785	Optical Disc archive, 4th gen., write once, 3300Gb
ODC5500R	786	Optical Disc archive, 5th gen., write once, 5500Gb
ECMA_322_1k	800	5,25", M.O., 4383356 sectors, 1024 bytes/sector, ECMA-322, ISO 22092, 9.1Gb/cart
ECMA_322_512	801	5,25", M.O., ??????? sectors, 512 bytes/sector, ECMA-322, ISO 22092, 9.1Gb/cart
ISO_14517	802	5,25", M.O., 1273011 sectors, 1024 bytes/sector, ISO 14517, 2.6Gb/cart
ISO_14517_512	803	5,25", M.O., 2244958 sectors, 512 bytes/sector, ISO 14517, 2.3Gb/cart
ISO_15041_512	804	3,5", M.O., 1041500 sectors, 512 bytes/sector, ISO 15041, 540Mb/cart
MetaFloppy_Mod_I	820	5.25", SS, DD, 35 tracks, 16 spt, 256 bytes/sector, MFM, 48 tpi, ???rpm
AtariLynxCard	821	
AtariJaguarCartridge	822	

Annex B. Data types

These are all the data types that can be contained in a data block or pointed by a deduplication table. They represent user data, media tags or sector tags.

Value	Data type
0	No data
1	User data
2	CompactDisc partial Table of Contents
3	CompactDisc session information
4	CompactDisc Table of Contents
5	CompactDisc Power Management Area
6	CompactDisc Absolute Time in Pregroove
7	CompactDisc Lead-in's CD-Text
8	DVD Physical Format Information
9	DVD Lead-in's Copyright Management Information
10	DVD Disc Key
11	DVD Burst Cutting Area
12	DVD DMI
13	DVD Media Identifier
14	DVD Media Key Block
15	DVD-RAM Disc Definition Structure
16	DVD-RAM Medium Status
17	DVD-RAM Spare Area Information
18	DVD-R RMD
19	DVD-R Pre-recorded Information
20	DVD-R Media Identifier
21	DVD-R Physical Format Information
22	DVD ADress In Pregroove
23	HD DVD Copy Protection Information
24	HD DVD Medium Status
25	DVD DL Layer Capacity
26	DVD DL Middle Zone Address
27	DVD DL Jump Interval Size
28	DVD DL Maual Layer Jump LBA
29	Blu-ray Disc Information
30	Blu-ray Burst Cutting Area
31	Blu-ray Disc Definition Structure

Value	Data type
32	Blu-ray Cartridge Status
33	Blu-ray Spare Area Information
34	AACS Volume Identifier
35	AACS Serial Number
36	AACS Media Identifier
37	AACS Media Key Block
38	AACS Data Keys
39	AACS LBA Extents
40	CPRM Media Key Block
41	Hybrid disc recognized layers
42	MMC Write Protection
43	MMC Disc Information
44	MMC Track Resources Information
45	MMC Pseudo-OverWrite Resources Information
46	SCSI INQUIRY response
47	SCSI MODE PAGE 2Ah
48	ATA IDENTIFY response
49	ATAPI IDENTIFY response
50	PCMCIA CIS
51	SecureDigital CID
52	SecureDigital CSD
53	SecureDigital SCR
54	SecureDigital OCR
55	MultiMediaCard CID
56	MultimediaCard CSD
57	MultiMediaCard OCR
58	MultiMediaCard Extended CSD
59	Xbox Security Sector
60	Floppy Lead-out
61	DVD Disc Control Block
62	CompactDisc First track negative pregap
63	CompactDisc Lead-out
64	SCSI MODE SENSE(6) response
65	SCSI MODE SENSE(10) response
66	USB descriptors
67	Xbox Disc Manufacturer Information
68	Xbox Physical Format Information

Value	Data type
69	CompactDisc sector prefix (sync, header)
70	CompactDisc sector suffix (edc, ecc p, ecc q)
71	CompactDisc subchannel
72	Apple Profile tag (20 bytes)
73	Apple Sony tag (12 bytes)
74	Priam Data Tower tag (24 bytes)
75	CompactDisc Media Catalogue Number
76	CompactDisc sector prefix (only incorrect ones stored)
77	CompactDisc sector suffix (only incorrect ones stored)
78	CompactDisc MODE 2 sector subheader
79	CompactDisc Lead-in
80	DVD Disc Key (decrypted)
81	DVD CPI_MAI
82	DVD Title Key (decrypted)

Annex C. Compression types

These are all compression types that can be used and found in AaruFormat images.

Value	Algorithm
0	None
1	LZMA. Stream prepended by 5 bytes of parameters.
2	FLAC.
3	LZMA after the data has been processed by the Claunia Subchannel Transform (see Annex D). Stream prepended by 5 bytes of parameters.

Annex D. Claunia Subchannel Transform

The subchannel in a CompactDisc media, and derived, consists of 8 elements: P, Q, R, S, T, U, V, W.

Usually they are interleaved, so each byte read from the media contains 1 single bit of each element.

This makes the data appear much more random than it really is, making LZMA be really slow an inefficient (less than 2% of compression gains) when applied to the data as it.

The transform is really simple, first all bits are de-interleaved so the 8 elements are individual bytes. Then all the P bytes from all the sectors are written sequentially before all Q bytes that are written before all R bytes and so on.

While this takes the double amount of memory for the transformation (approximately 32MiB more of memory), the resulting data is compressed up to 10 times faster with a compression gain of approximately 96% (on discs that do not contain data in the R-W subchannels, as 99% of discs are).