Disk Image Content Model and Metadata Analysis

ACTIVITY 1: Comparative Format Matrix Analysis

DRAFT - 2016-02-25

Submitted By:

253 36th Street, Suite C302
Brooklyn, NY 11232

Telephone 1.917.475.9630

bertram@AVPS.com

February 2016

Table of Contents

- Introduction
- Notes about Families of Disk Image Formats
- Formats Evaluated
- Analysis Process
- Analysis Scoring
  - Additional Considerations
    - Compression
    - Raw Capture
    - Capture Targets
      - Physical capture
      - Logical capture
      - Selected files
    - Optical capture
    - Metadata
    - Splitting
    - Error Information
    - Formats Within Formats
- Analysis Outcome
  - Format Scenarios
  - Rejected Formats
- Class A Format Profiles

Introduction

Creating disk images is a common practice in computing, with applications in software deployment, drive and tape backups, and law enforcement investigative analysis, among many others. In the library and archive sectors, the practice of generating disk images has found favor as a method of digital preservation that supports “reformatting” tangible media carriers (e.g., hard drives, floppy disks, SD cards, flash drives, ZIP disks, optical discs) so that the bits stored on the magnetic or optical carriers can be stored as carrier-independent files and managed within controlled digital preservation storage environments. Concomitant with this dematerialization of bits is the necessity of retaining the existing order in which the bits were stored and understood on the original tangible media carrier. The process of disk imaging is the process of serialization, i.e., the process of reading bytes from an original storage device sector-by-sector and writing copies of those bytes into a new, addressable sequence of bytes (a.k.a., a file, or, in some cases, a set of files).

How these sequences are constructed into files is where differences in disk image formats emerge. Some applications of disk imaging require only a raw byte-by-byte copy of the original bytes stored on the original device, from $byte_0$ to $byte_n$. Other use cases require added structure around the copied sectors to ensure integrity of the copy (e.g., CRCs after each sector) or additional metadata to document provenance (e.g., headers that capture a law enforcement case number or the investigator’s name). Certain use cases require the addition of sidecar text files to document the offsets where certain segments begin and end within a sequence of bytes.

The goal of this activity is not to recommend a single preferred disk image format for all preservation activities at HL, but rather to characterize and compare a set of viable formats widely in use in disk imaging applications—generally areas where the digital product includes a serialized sector-by-sector copy of bytes stored on a tangible media carrier. The output of this exercise provides a resource that can be used to compare and contrast the various attributes, characteristics, advantages, and disadvantages of each format, ultimately to assist in establishing Class A (Preferred) and Class B (Accepted) sets of formats that will be supported within HL’s DRS. Although a wider variety of disk image formats could be compared, we analyzed a subset that represent formats commonly used in archive and library settings, as well as a few old and new formats that are not so widely employed but warranted consideration (because either they may be in existence currently or because it is likely the format will gain popularity in the upcoming years).

Notes about Families of Disk Image Formats

Throughout this process, we reviewed the following formats as part a comparative analysis of disk image format sustainability, including the following format groups:

- Magnetic Carrier
    - Raw: DD, IMG, RAW
    - Formatted: E01, L01, Ex01, Lx01, S01, AFFv1-3, AFF4
- Optical Carrier
    - Raw: BIN/CUE
    - Formatted: ISO 9660

We considered DMG, but found that this format is more closely related to standard serialization formats (ZIP, TAR, etc.), which, in reality are used in a slightly different set of applications than what would be considered “disk imaging” applications.

As one focuses on the actual ability (or not) that a repository would have in trying to preserve and provide access to these formats (and the content stored therein) over time, the underlying file system becomes the biggest hurdle. This information is not something addressed very well by these formats. As will become clear in the metadata analysis phase of this project, DFXML does a useful job in attempting to solve that dilemma—by providing a structured way to capture information about the included file system formats in a given image.

Formats Evaluated

The following formats were selected for this comparison project:

1. EWF-E01. Expert Witness Compression Format. This format, as a proprietary format of EnCase and ASR Data has been basically deprecated, however, the open-source community has reverse engineered the format and published a stable specification that is used in libewf software libraries. It is designed to support arbitrary offset access within compressed data streams and it uses CRC's for integrity validation. This format is basically the same as EWCF-ASR02. EnCase recently moved to a new version of this format -- EWF2-Ex01. There are small differences internally between EWF-ASR, EWF-SMART, EWF-E01 (and the various implementations in EnCase versions and FTK versions), and the EWF-L01 implementations. These differences are documented in the Metz version of the specification.
    a. Lineage:
        i. Expert Witness Format - ASR Data Format (ASR02)
        ii. EnCase Formats v1,2,3,4,5,6,7
        iii. Expert Witness Compression Format, version 1 specification, maintained by Joachim Metz
2. EWF-L01. Logical Evidence File, introduced in EnCase 5, also stored in EWF format.
    a. Lineage:
        i. Logical Evidence File (LVF) - EnCase 5
3. EWF2-Ex01. Encase Evidence File Format Version 2. Many of the central design principles of the E01 format have been retained. The Ex01 format still stores data in blocks that are verified with an individual 32-bit CRC, and all of the source data stored in the file is hashed with the MD5 and/or SHA-1 algorithms if requested by the user. The Ex01 enhancements do not affect features of the file such as these that many courts have relied on to rule that the file is an accepted container of original evidence; the additions merely facilitate the ability to track and handle new characteristics of the stored data. The new Ex01 format introduces the following capabilities: Support for encryption of the data; Ability to use different compression algorithms; Improved support for multi-threaded acquisitions, where sectors can be out of order; Efficient storage and handling of sector blocks that are filled with the same pattern (such as 00-byte fills); Alignment considerations to improve efficiency and performance; Improved support for resuming acquisitions; Internal improvements of the data structures. While some of this new functionality is not yet fully leveraged in the current version, all necessary data is stored, the

data structures support expansions, and subsequent versions will use this new format to its fullest.

   a. Lineage:
      i. EnCase Formats v1,2,3,4,5,6,7
      ii. Expert Witness Compression Format, version 2 specification, maintained by Guidance Software
      iii. Expert Witness Compression Format, version 2 specification, maintained by Joachim Metz

4. EWF2-Lx01. Logical Evidence File, introduced in EnCase 7, also stored in EWF format.
   a. Lineage:
      i. Logical Evidence File (LVF) - EnCase 5
      ii. Expert Witness Compression Format, version 2 specification, maintained by Guidance Software
      iii. Expert Witness Compression Format, version 2 specification, maintained by Joachim Metz
5. Smart S01. SMART format - SMART is a software utility for Linux designed by the original authors of Expert Witness (now sold under the name EnCase). It can store disk images as pure bitstreams (compressed or uncompressed) or in ASR Data's Expert Witness Compression Format. Images in the latter format can be stored as a single file or in multiple segment files, consisting of a standard 13-byte header followed by a series of sections, each of type "header," "volume," "table," "next" or "done." Each section includes its type string, a 64-bit offset to the next section, its 64-bit size, padding, and a CRC, in addition to actual data or comments, if applicable. Although the format's "header" section supports free-form notes, an image can have only one such section (in its first segment file only).
   a. Lineage:
      i. Based on Expert Witness Format - ASR Data Format (ASR02)
      ii. ASR Data Smart Format (specification not published)
6. AFFv1-3. Deprecated. The Advanced Forensic Format is designed as an alternative to current proprietary disk image formats. AFF offers two significant benefits. First, it is more extensible because it allows extensive metadata to be stored with images. Second, AFF images consume less disk space than images in other formats (e.g., EnCase images). Developed by S. Garfinkel, D. Malan, K. Dubec, C. Stevens, and C. Pham in 2005.
   a. Lineage:
      i. AFFv1-3 - Advanced Forensic Format: An Open, Extensible Format For Disk Imaging (2005)
7. AFF4. RDF-based imaging; Object-oriented architecture; parts are stored as objects (streams, volumes) and assigned URNs that can be used to assert facts about the objects. Less of a file format and more of a complete evidence management system. Format developed by Michael Cohen, Simson Garfinkel, and Bradley Schatz.
   a. Lineage:
      i. AFF4 - Extending the advanced forensic format to accommodate multiple data sources, logical evidence, arbitrary information and forensic workflow (2009)
      ii. AFF4 - Wirespeed: Extending the AFF4 forensic container format for scalable acquisition and live analysis (2015)
8. RAW (DD, IMG, RAW). These formats represent unstructured byte sequences captured from physical or logical volumes. Many extensions have been used in the past when raw disk images were created. .raw, .dd, and .img are some of the more popular in our cultural heritage sector.

These represent any image that is solely an uncompressed sector-by-sector copy of stored bits (with no additional headers/footers/structure/metadata).

9. ISO. Optical media disk images. File structure of internal optical disc file system - ISO 9660 for CDs, UDF for DVDs. An ISO disk image, stores only file system information (not all bytes on all sectors, only user portion of 2,048 bytes per sector).
    a. ISO 9660:1988. Information processing -- Volume and file structure of CD-ROM for information interchange
    b. ISO/IEC 13346, parts 1 through 5, 1995 and 1999; running title Information technology -- Volume and file structure of write-once and rewritable media using non-sequential recording for information interchange; see list under Specifications below.
    c. ECMA-167. Volume and File Structure for Write-Once and Rewritable Media using Non-Sequential Recording for Information Interchange, 3rd edition (June 1997).
    d. IEEE P1282, Rock Ridge Interchange Protocol Draft Standard, version 1.12, adopted 1994-07-08.
    e. Joliet Specification, Extensions to the CD-ROM Recording Spec ISO 9660:1988 for Unicode, Version 1; May 22, 1995.
10. BIN/CUE. Optical media disk image. Raw disk information (BIN) plus metadata as text log of track separations within disk image (CUE). Bin files contain all bytes on all sectors - all 2,352 bytes of each sector including control headers and error correction data. Original CUE metadata specification is published in CDRWIN documentation, 2000.
    a. Lineage:
        i. CDRWIN documentation, Chapter 7, 2000.

Analysis Process

The goal of this format analysis is to characterize and compare a set of viable formats widely available in the current environment disk imaging — generally areas where the digital product includes a purposefully created sector-by-sector copy of a physical storage device or a logical directory-by-directory copy of a specified filesystem. The output of this exercise will provide a resource that can be used to compare and contrast the various attributes, characteristics, advantages, and disadvantages of each format, ultimately to assist in establishing Class A (Preferred) and Class B (Accepted) sets of formats that will be supported within HL's DRS.

Following in the footsteps of previous Federal Agencies Digitization Guidelines Initiative (FADGI) comparative analyses[1], we did not attempt to apply precise rankings for each factor that we assessed across each format. Instead, we established general three-level scoring (bad, good, better) for each criterion, documenting the methodology used to score a given criterion, as well as articulating how the score was determined. We borrowed terminology from the Library of Congress' format sustainability website.

Reviewing format specifications (when possible), publications, and the files themselves, each format was evaluated according to a set of 29 specific characteristics (listed below). These characteristics were grouped into the above-mentioned FADGI groupings: Sustainability Factors, Cost Factors, and System Implementation Factors.

[1] Analyses available here: http://www.digitizationguidelines.gov/guidelines/File_format_compare.html.

FADGI Grouping	Specific Characteristic
Sustainability Factors: Adoption	Dependency on a single organization or company
Sustainability Factors: Adoption	Widespread use by consumers
Sustainability Factors: Adoption	Widespread use by professionals
Sustainability Factors: Adoption	Archival use
Sustainability Factors: Adoption	Community / 3rd party support
Sustainability Factors: Adoption	Developer / corporate support
Sustainability Factors: Adoption	Geographic spread
Sustainability Factors: Adoption	Lifetime
Sustainability Factors: Adoption	Revision rate
Sustainability Factors: Adoption	Last revision
Sustainability Factors: Disclosure	Availability of specifications
Sustainability Factors: Disclosure	Degree to which specification is complete and understandable
Sustainability Factors: Disclosure	Standardized
Sustainability Factors: Impact of patents	Legal restrictions affecting use now or long-term
Sustainability Factors: Native Embedded Metadata Capabilities / Settings and Maintenance: support for color maintenance	Technical metadata support
Sustainability Factors: Technical Protection Mechanisms	Support for technical protection mechanisms
Sustainability Factors: Transparency	Level of format complexity
Sustainability Factors: Transparency	Degree to which compression is understood
Sustainability: Native Embedded Metadata Capabilities	Descriptive metadata support
Cost Factors: Cost of providing access	Browser support
Cost Factors: Implementation Costs	Cost to maintain environment for access or processing
System Implementation Factors: Availability of Tools	Dependencies on particular HW/SW
System Implementation Factors: Availability of Tools	Quantity and availability of mounting software
System Implementation Factors: Ease and accuracy of file validation	Ease of accurate identification
System Implementation Factors: ease and accuracy of file validation	Ease of accurate validation
System Implementation Factors: Ease and accuracy of monitoring quality	Difficulty of tool dev for accurate identification
System Implementation Factors: Level of Difficulty	Format expertise available

System Implementation Factors: Level of Difficulty	Backward compatibility
System Implementation Factors: technical complexity	Error-tolerance

Table 1. Characteristics evaluated for each format.

The full and final matrix is available at the following link:
https://docs.google.com/spreadsheets/d/1yPrTHYLfXfp0r-tSjni33YaFOBnmi4wV-u2zj9fhktU/edit?usp=sharing

Analysis Scoring

In order to produce a quantitative result from the 29 characteristics evaluated for each format, each of the above evaluated characteristics was graded on a scale of 1-3 (from worst to best). From these broad scores, we counted the low, medium, and high scores for each format and produced a weighted overall score for the format. The calculation is as follows:

weighted score = [# of high scores]*1 + [# of medium scores]*0.5 + [# of low scores]*0.25

Additional consideration was given to formats that scored low generally but that are currently employed as preservation targets by repositories, or in cases where a format is so ubiquitous that it would be a hindrance to HL's mission to reject the format entirely.

Two additional weightings were applied to these weighted scores: using a grouping of characteristics previously applied by HL—where 3 sets of characteristics were graded as less important, important, and most important. Again we used values of 1, 2, and 3 respectively. When a characteristic received an initial weighted score of 3 if the particular characteristic was graded as "most important" by HL, then the field received an actual value of 3 x 3, or 9. Scores were then aggregated per format to deliver a total format score using the HL weighting.

Similarly, AVPreserve applied a preservation weighting to the characteristics that rated the characteristics in light of concerns for long-term preservation (this is similar to the HL weighting, but from the perspective of AVPreserve staff members). If a characteristic was deemed to be absolutely critical for long-term care and management then it received a value of 3. Characteristics that were important but not critical for long-term care and management received a value of 2; and those that had the least effect on long-term care received a 1. These weightings were then scored in the same way that the HL weightings were scored.

A matrix of the final scores follows, organized from highest to lowest score in the "HL Color Coded Weighting" column.

	Standard Weighted Score	**HL Color Coded Weighting**	**AVPreserve Weighting**
RAW (IMG, DD)	24.5	51	56.5
ISO	21.75	44.5	49.25

BIN/CUE	20.5	41.75	49.25
EWF-E01 (EWCF-ASR02)	19.25	38.5	45.25
AFFv1-3 (AFF, AFD, AFM)	17.25	35.75	42
EWF-L01	17.75	35	43.25
S01 (.001)	16	32.25	38.75
AFF4	14.75	31.25	33.75
EWF2-Lx01	14.5	29.5	36.5
EWF2-Ex01	13.75	27.75	33.75

Additional Considerations

The following considerations are included here as issues that HL should consider in light of disk image format management. These factors were all considered in line with the results of the comparative format matrix in order to determine the following recommendations for Class A and Class B image sequence formats.

Compression

Disk images are typically employed to create file-based copies of physical storage devices. Most physical storage devices are made to exploit storage capacity to the fullest, so we find that most disk images of modern day storage devices yield very large individual files. Because of this filesize dilemma, many creators of disk images desire to use compression strategies to minimize filesize. In almost every case of disk image compression, lossless strategies are used (as opposed to lossless compression techniques that we find in media files -- where often a goal is to deliver the highest quality media in the smallest package possible). In disk imaging activities, strategies for compression come mainly in two flavors. When raw image formats are used, people employ compression via gzip compression (on top of .tar files), or similar lossless zip compression techniques. The file is captured raw, and then secondary compression is applied to the entire file after the capture. The second approach is found in many disk image formats that come out of the forensic use-cases. In this approach, zlib (or similarly lossless) compression techniques are applied to sections within the file to optimize compression opportunities in smaller chunks, compounding the total compression of the file and providing for even smaller disk image files. Although both approaches employ lossless (and relatively well-known and well-documented) compression techniques, one compresses as part of the file format (a more complex approach in terms of sustainability) and the other compresses as a secondary wrapper to a raw format.

For example, the EnCase/ASR Expert Witness Format, E01, can capture physical devices, logical drives, and individually specified files (L01) while writing bytes to the disk image file format. This format offers users the options to store the bytes with compression (zlib) or as uncompressed data. When compressed, only structured parts of the file (such as the original header) are stored uncompressed. All other metadata and file data is stored in zlib compression (depending on the quality assigned by the user -- anywhere from 1 to 9, in some unspecified spectrum of compression quality). Data cannot be read as easily with a hex editor in these cases. Data must be uncompressed before it can be understood. Organizations that support E01 formats should ensure that they store and maintain the ability to decompress with the zlib compression library. When files are stored uncompressed, the data is in full view with hex editors. The only downside is that the uncompressed file can be so large that no hex editor can hold the file in memory. In these cases, the file must be paged through using a command line hex dump.

Raw Capture

Because so many tools can expose the stored bytes of a raw format, this is the de facto standard in the disk imaging community. When utilizing raw formats, it is imperative to capture information about the format of the file system(s) and the original device being captured so that the raw binary data can be deciphered appropriately in the future.

Capture Targets

Raw disk image formats and structured disk image formats (i.e., "forensic" formats), can target information on a physical device from three perspectives. They can treat the device as faceless bit-bucket and read the bytes sector-by-sector, capturing every byte on the physical storage device exactly as written on the device. This is a physical capture. Another approach is to read the file system(s) on the storage device and to select a file system to capture—from start to finish, byte-for-byte of the logical file system. In this case, if other file systems exist on the physical device, or if other data not part of the file system exists on the device, those bytes will not be captured. This is a logical capture. The last approach is even more restricting. In this case a user selects a set of files or directories within a file system on a physical device and asks the disk image utility to extract only the bytes that constitute those files. We see this only in structured disk image formats, and usually used to attach evidentiary information to the extraction process so that the files can be examined as evidence and held up in court. This is a capture of selected files.

Physical capture

Tools, such as FTK Imager or dd, can write the bytes off of any device that the host computer has access to. A physical disk image capture scrapes all bytes from a physical device regardless of file systems and volumes contained on the device. Nothing on the device is augmented, but future access to the new disk image is going to be heavily dependent on knowing what file systems and volumes were originally contained on the device. An archive will need to capture this information separately and maintain the ability to mount the contained file systems moving forward. Brute force scans, such as bulk_extractor will be able to perform reg_ex evaluations of the content in order to report a variety of metadata.

Logical capture

A logical capture focuses on specific volumes as mounted by the host computer. In this case, only the content identified by the file system to be part of the file system is contained in the resulting disk image file.

Selected files

A logical capture can also focus on specific files and directories on volumes mounted by the host computer. In this case, only the selected content is captured in the resulting disk image file.

Optical capture

Optical disk sectors can be written to a raw image, just as any physical device can be. In these cases, all control headers and error correction fields (from CD-ROM or DVD) are included in the image because the image reads every byte on the disk. This is another example of a physical capture. The common file system of an optical disk (ISO 9660 - for CDs or UDF - for DVDs) can also be captured as a logical volume. In these cases, typically called ISO images, only the file system is captured from the optical media (leaving behind any bytes not included within the file system). As we would expect to see with a logical capture.

The important question here is whether an archive is capturing a raw version of a file system (in which case the archive will need to be able to support access to that file system in order to get access to the files within it), or is the archive capturing a raw version of a physical device (in which case the archive will need to know which file system(s) exist within the device and how they are arranged so that the archive can get access to the files contained within the file systems).

Metadata

Most structured disk image (read: non-raw) formats claim that the benefit of structure is the ability to store metadata about the device/volume being imaged. Metadata can be embedded within the image file in these cases. Other approaches, used with both raw and structured formats, is to store metadata externally by employing schemas such as DFXML or CUE.

Splitting

Most disk image formats can be split into smaller segment files because of the relative size of most devices being imaged, especially when the resulting disk image files are stored in a FAT32 file system where size limits are at 2GB/file. E01/S01 have specifications for handling split image files, using the extension with one-up numbering sequences. AFF requires the use of AFD for splitting. SMART default (raw with external metadata) uses .001 sequentially. Contemporary implementations of E01 (using the libewf library) are attempting to support larger file sizes to minimize the need for split files.

If splitting is necessary for a certain capture, an archive should consider policies and procedures for ensuring the files are kept together and associated metadata is maintained and associated with the appropriate files in the sequence.

Error Information

Commonly, disk image software writes 0’s in place of data that cannot be read from a physical storage device during capture. In raw images, because of the lack of metadata to document the decisions of the software during capture, this makes it difficult to identify bad sectors (a sector with all 0’s) versus sectors that originally stored all 0’s. Structured disk image formats (E01, AFF) make it possible for the capture software to store process history information within the file so that it becomes possible for an examiner to understand if bad sectors exist in the image, and, if so, where.

Formats Within Formats

Disk images can be a valuable way to stall for time, in that they take media deterioration off the list of risks by shifting the bits wholesale from disks to server-ish carriage. But the image files do not in and of themselves address the problem of content viability over time. They kick the can down the road. An archive must make efforts to understand the formats stored within disk images so that these formats can benefit from obsolescence monitoring and format migration as necessary to ensure continued understandability.

Analysis Outcome

A matrix of the scores and final classifications follows. Green rows indicate our recommendations for Class A formats based on the analysis completed in this phase of the disk image content modeling project (these selections were determined in accordance with HL team members). AVPreserve and HL team members agreed that there is no need for B formats in this case, preferring to support only one "forensic" use-case format at this point. Red rows indicate the formats that were not considered stable enough for inclusion in the HL DRS at this point. Because of the slight instability in the forensic format environment, AVPreserve recommends a continued analysis of the changes in this area over the next five years.

	Standard Weighted Score	**HL Color Coded Weighting**	**AVPreserve Weighting**	**Classification**
RAW (IMG, DD)	24.5	51	56.5	A
ISO	21.75	44.5	49.25	A
BIN/CUE	20.5	41.75	49.25	A
EWF-E01 (EWCF-ASR02)	19.25	38.5	45.25	A
AFFv1-3 (AFF, AFD, AFM)	*17.25*	*35.75*	*42*	
EWF-L01	*17.75*	*35*	*43.25*	
S01 (.001)	*16*	*32.25*	*38.75*	
AFF4	*14.75*	*31.25*	*33.75*	
EWF2-Lx01	*14.5*	*29.5*	*36.5*	
EWF2-Ex01	*13.75*	*27.75*	*33.75*	

Format Scenarios

Based on the format assessment, following are example scenarios where HL could employ our recommended classifications.

- Magnetic storage device - physical sector-by-sector copy
    - Class A
        - Uncompressed
            - RAW (IMG,DD)
                - Unless file size is above certain limit (e.g., 1 TB)
        - Compressed
            - RAW (IMG,DD) with added TAR.GZ or ZIP
            - E01
                - Uses zlib compression option

- Magnetic storage device - logical file system copy
    - Class A
        - Uncompressed

            - RAW (IMG,DD)
                - Unless file size is above certain limit (e.g., 1 TB)
        - Compressed
            - RAW (IMG,DD) with added TAR.GZ or ZIP
            - E01
                - Use zlib compression option

- Optical storage device - physical sector-by-sector copy
    - Class A
        - Uncompressed
            - BIN/CUE

- Optical storage device - logical file system copy
    - Class A
        - Uncompressed
            - ISO

Rejected Formats

- AFFv1-3
    - migrate to E01
    - use libewf or FTK
- EWF L01
    - extract and migrate to ZIP or BagIt
- S01
    - migrate to E01
    - use libewf or FTK
- AFF4
- EWF2 Ex01
- EWF2 Lx01

Class A Format Profiles

Class A

RAW (IMG, DD)	
Full name	Raw disk image
Description	These formats represent unstructured byte sequences captured from physical or logical volumes. Many extensions have been used in the past when raw disk images were created. .raw, .dd, and .img are some of the more popular in our cultural heritage sector. These represent any image that is solely an uncompressed sector-by-sector copy of stored bits (with no additional headers/footers/structure/metadata).
Class	A (Magnetic media)
Key adopters	Used globally in disk image applications for more than 20 years. Raw disk images are the de facto standard for disk image applications.
Applicable MIME media types	application/octet-stream
Applicable file extensions	.raw, .dd, .img
Original developer	Not managed by a single developer
Current developer	Not managed by a single developer
Specifications (URL in parenthesis)	There is no format specification for raw files. They are sector-by-sector copies of a physical device or a segment of a physical device.
Patent/license issues	None.
Key related links	http://www.forensicswiki.org/wiki/Raw_Image_Format https://en.wikipedia.org/wiki/Disk_image http://forensicswiki.org/wiki/Dd https://en.wikipedia.org/wiki/IMG_(file_format)

E01	
Full name	Expert Witness Compression Format - EWF-E01
Description	This format, as a proprietary format of EnCase and ASR Data has been basically deprecated, however, the open-source community has reverse engineered the format and published a stable specification that is used in libewf software libraries. It is designed to support arbitrary offset access within compressed data streams and it uses CRC's for integrity validation. This format is basically the same as EWCF-ASR02. EnCase recently moved to a new version of this format -- EWF2-Ex01. There are small differences internally between EWF-ASR, EWF-SMART, EWF-E01 (and the various implementations in EnCase versions and FTK versions), and the EWF-L01

	implementations. These differences are documented in the Metz version of the specification.
Class	A (Magnetic media)
Key adopters	Law enforcement evidence examiners; digital humanities researchers; manuscript archives
Applicable MIME media types	application/octet-stream
Applicable file extensions	.E01, .E02, .Exx
Original developer	ASR Data
Current developer	Guidance Software, Inc.; open-source libewf community
Specifications (URL in parenthesis)	Expert Witness Format - ASR Data Format (ASR02) (http://forensicswiki.org/wiki/ASR_Data's_Expert_Witness_Compression_Format) EnCase Formats v1,2,3,4,5,6,7 (not publicly available) Expert Witness Compression Format, version 1 specification, maintained by Joachim Metz (https://53efc0a7187d0baa489ee347026b8278fe4020f6.googledrive.com/host/0B3fBvzttpiiSMTdoaVExWWNsRjg/Expert%20Witness%20Compression%20Format%20(EWF).pdf)
Patent/license issues	Developed by ASR Data and Guidance Software. Not an open format. Existing public documentation is Copyright Joachim Metz and made available under a GNU Free Documentation License, Version 1.3.
Key related links	http://www.forensicswiki.org/wiki/Encase_image_file_format http://www.digitalpreservation.gov/formats/fdd/fdd000408.shtml http://www.digitalpreservation.gov/formats/fdd/fdd000406.shtml

ISO	
Full name	ISO 9660 Disk Image Format
Description	File structure of internal optical disc file system - ISO 9660 for CDs, UDF for DVDs. An ISO disk image, stores only file system information (not all bytes on all sectors, only user portion of 2,048 bytes per sector).
Class	A (Optical discs)
Key adopters	Used globally in optical disc imaging.
Applicable MIME media types	application/x-iso9660-image
Applicable file extensions	.iso
Original developer	International Organization for Standardization (ISO)
Current developer	International Organization for Standardization (ISO)

Specifications (URL in parenthesis)	There is no comprehensive single specification for all of the variant formats called ISO image. Three international standards offer specifications for the relevant disk formats and, by implication, for the media-independent representation of disk content. Other relevant specifications have been published by standards bodies and corporate entities. ISO 9660:1988. Information processing -- Volume and file structure of CD-ROM for information interchange. ISO/IEC 13346, parts 1 through 5, 1995 and 1999; running title Information technology -- Volume and file structure of write-once and rewritable media using non-sequential recording for information interchange. ECMA-167. Volume and File Structure for Write-Once and Rewritable Media using Non-Sequential Recording for Information Interchange, 3rd edition (June 1997). IEEE P1282, Rock Ridge Interchange Protocol Draft Standard, version 1.12, adopted 1994-07-08. Joliet Specification, Extensions to the CD-ROM Recording Spec ISO 9660:1988 for Unicode, Version 1; May 22, 1995.
Patent/license issues	No known issues.
Key related links	https://en.wikipedia.org/wiki/ISO_image https://en.wikipedia.org/wiki/ISO_9660 http://www.digitalpreservation.gov/formats/fdd/fdd000348.shtml http://www.avpreserve.com/wp-content/uploads/2014/04/OpticalMediaPreservation.pdf

BIN/CUE	
Full name	BIN/CUE Binary and Metadata Disk Image
Description	Optical media disk image. Raw disk information (BIN) plus metadata as text log of track separations within disk image (CUE). Bin files contain all bytes on all sectors - all 2,352 bytes of each sector including control headers and error correction data. Original CUE metadata specification is published in CDRWIN documentation, 2000.
Class	A (Optical discs)
Key adopters	Used globally in optical disc imaging for over 20 years.
Applicable MIME media types	application/octet-stream; application/x-binary
Applicable file extensions	.bin, .cue
Original developer	Jeff Arnold; CDRWIN
Current developer	Not currently maintained

Specifications (URL in parenthesis)	CDRWIN documentation, Chapter 7, 2000. (http://wiki.hydrogenaud.io/index.php?title=Cue_sheet) The Red Book is the colloquial name for IEC 60908, due to the color of its cover. The various standards that define optical media are collectively known as the Rainbow Books, as each has a differently colored cover (e.g. ISO/IEC10149 being the Yellow Book).
Patent/license issues	No known issues.
Key related links	http://www.avpreserve.com/wp-content/uploads/2014/04/OpticalMediaPreservation.pdf https://en.wikipedia.org/wiki/Cue_sheet_(computing)