



University
of Glasgow

Kim, Y., and Ross, S. (2011) *Digital forensics formats: seeking a digital preservation storage format for web archiving*. In: International Digital Curation Conference (IDCC 2011), 5-7 December 2011, Bristol, UK.

<http://eprints.gla.ac.uk/62565/>

Deposited on: 21th May 2011

7th International Digital Curation Conference

December 2011

Digital forensics formats: seeking a digital preservation storage format for web archiving

Yunhyong Kim,

Humanities Advanced Technology and Information Institute,
University of Glasgow, Glasgow, UK.

Seamus Ross,

Faculty of Information,
University of Toronto, Toronto, Canada.

&

Humanities Advanced Technology and Information Institute,
University of Glasgow, Glasgow, UK.

<month><20xx>

Abstract

In this paper we discuss archival storage formats from the point of view of digital curation and preservation. Considering established approaches to data management as our jumping off point, we selected seven format attributes which are core to the long term accessibility of digital materials. These we have labeled core preservation attributes. These attributes are then used as evaluation criteria to compare file formats belonging to five common categories: formats for archiving selected content (e.g. tar, WARC), disk image formats that capture data for recovery or installation (partimage, dd raw image), these two types combined with a selected compression algorithm (e.g. tar+gzip), formats that combine packing and compression (e.g. 7-zip), and forensic file formats for data analysis in criminal investigations (e.g. aff, Advanced Forensic File format). We present a general discussion of the file format landscape in terms of the attributes we discuss, and make a direct comparison between the three most promising archival formats: tar, WARC, and aff. We conclude by suggesting the next steps to take the research forward and to validate the observations we have made.



1. Introduction

The selection of a storage format for digital material that facilitates the long-term accessibility to digital object content, and supports the continuation of behaviour and functionalities associated to digital objects, is one of many core tasks of a digital archive. This task is especially challenging with respect to complex aggregate digital objects such as weblogs, involving multimedia objects that are produced in varying formats to carry out a wide range of interactive functionalities, including dynamic changes overtime, and displayed using distributed information within the context of social networks. As a first step to meet this challenge, we present here results of our preliminary investigations examining storage formats likely to benefit a dynamic weblog archive, a study conducted as part of the BlogForever project¹, aiming to create a platform for aggregating, preserving, managing and disseminating blogs.

There have been many studies on the impact of digital object formats on the preservation of digital objects (e.g. Brown (2008); Todd (2009); Buckley (2008); Christensen (2004); Fanning (2008); McLellan (2006)). The retention of essential object properties can be facilitated by examining the preservation attributes of the file format. Some of these (e.g. scale of adoption and disclosure, support for data validation, and flexibility in embedding metadata) have surfaced elsewhere as sustainability factors (cf. Library of Congress sustainability factors²; Arms & Fleischhauer (2003); Rog and van Wijk (2008); Brown (2008)) and capacity of the format to retain significant digital object properties (Hedstrom & Lee (2002); Dappert & Farquhar (2009); Guttenbrunner et al. (2010)).

Most of these studies, however, seem to be focused on considerations of individual digital object formats, and, even then, generate many differences of opinion. There has been little consensus on best practices for selecting storage container formats (e.g. tar) that aggregate or capture collections composed of several object types, such as we might encounter within a web archiving environment. While formats such as WARC [A3]³ have been proposed and developed into an international ISO⁴ standard, these recommendations are rarely based on a comparison of a range of formats using the full range of preservation attributes within the same environmental setup. Even when storage architecture is discussed on a wider scale, it often comes focused on one or two selected factors⁵ (e.g. software and hardware scalability and costs).

In the following, we discuss a core set of preservation attributes for storage formats. These include those that have been addressed in common by several previous studies on file formats (e.g. conducted by the UK Digital Curation Centre (e.g. Abrams (2007), the US Library of Congress, and the technology watch reports published by the Digital Preservation Coalition (e.g. Todd (2009))). The set has, however, been augmented by an increased cognizance of the concept the quality and completeness of

¹funded by the European Union's Seventh Framework Programme (FP7-ICT-2009-6) under grant agreement n° 269963

²http://www.digitalpreservation.gov/formats/intro/format_eval_rel.shtml

³ Throughout this paper, references in square brackets, expressed as a number preceded by the letter A, refer to those in the left most column of the table in “Appendix I: table of archival formats and compression utilities”, included at the end of this paper.

⁴<http://www.iso.org>

⁵http://www.digitalpreservation.gov/news/events/other_meetings/storage10/

data. The ability to represent the full digital content of an object and or data has been observed as a relevant factor before (e.g. Todd (2009); Pipino, Lee & Wang (2002); Batini & Scannapieco (2006); Huc et al. (2004)), however, the “completeness of data” we address here refers to much more than the target digital object content. For example, in the digital environment, provenient evidence surrounding digital objects can be derived from recorded file modification dates, lists of files that were deleted, and logs of processes and resulting errors, and logs of programs that had been run on the system. This kind of history is retained on the system disk, and, ideally, as a standard practice in systems administration, should be retained to trace accountability (not only with respect to humans but also software and hardware). Once you reduce the preservation activity to that associated digital objects only, all this supporting information tends to become hidden and may be, even, lost.

We have also placed more emphasis on scalability (e.g. measured by compression ratio to meet storage requirements, and decompression speed to reduce overheads on any processes that take place on the material) and flexibility (e.g. being able to deal with multiple types, sizes, and numbers of digital material through a variety of operating systems). The scalability and flexibility is crucial within the web environment where we need to support rapidly growing data, distributed processing, aggregation of multimedia objects, and sophisticated approaches to search.

In the next section, these observations will be reflected in our proposal of seven core attributes for storage formats. We will then discuss a range of formats (see Section 3 and Appendix I) with respect to these attributes, and make some concluding remarks with suggestions of next steps in the final section.

2. Seven core preservation attributes for file formats

In this section, we propose seven core attributes that should be assessed with respect to storage container formats for the purposes of supporting digital preservation. As mentioned in the previous section, these attributes were selected to reflect preservation requirements identified through other research and application development initiatives, e.g. the sustainability factors for formats discussed at the Library Congress⁶.

Previous studies, however, have placed much concentration on front-end isolated formats for individual digital objects. The attributes here include the notion of completeness of data intended to consider the extent of contextual data (e.g. file system information, permissions, and error logs) surrounding the object that is being captured. We also put emphasis scalability, not only in terms of minimising storage, and optimising management efficiency with respect to variations in the quantities of data (e.g., crucial in the case of web archives that become increasingly bigger in size and diverse with respect to included object types, or data collected from scientific instruments), but, also in terms of reducing overhead with respect to sophisticated data mining and search technologies that are likely to play a bigger role in the future. The attributes are described below along with Library of Congress sustainability factors (LC SF) in parenthesis, for comparison, where relevant:

⁶<http://www.digitalpreservation.gov/formats/sustain/sustain.shtml>

4 Digital forensics formats

←—————→

1. **Completeness of data:** the format should preserve data as closely as possible to a sector-by-sector copy of the raw data on a system disk, i.e. inclusive of file structure, dependencies, and history. This:

- minimises deterioration and information loss.
- maximises the chances of preserving file system information (directory structure, file size, permissions, encoding, any relationships and dependencies between files and executables).
- increases the possibility of retaining extra information about changes that have been made on the disk to be used for tracking accountability, integrity, authenticity, and maximising recoverability (see also 2 and 3).

2. **Recoverability of data:** the format should support the recovery of data wherever possible, e.g. one corrupted file or sector, if possible, should not pose serious problems in recovering other files or sectors in the archive.

3. **Support for data validation** (cf. LC SF “technical protection mechanisms”): the format should support validation procedures. For example, the format should support:

- piecewise hash codes and digital signatures to verify it as an authentic original copy, and,
 - provide means of encryption to protect the data from manipulation.
- While these functions can be added in some cases, it is best to minimise the accumulation of third-party tools and added procedures as this increases overhead (see discussion of scalability in 4) and the margin for introducing errors.

4. **Scalability of data management processes:** the format should have properties that make all processes within the archive scalable to handle files of any size, datasets of any size and added services. In particular, the format should:

- not limit the size of input file, output file, and/or media.
- support efficiency with respect to storage and processing speed, e.g. the format should 1) use effective compression methods to reduce storage requirements, 2) if possible, not require decompression for searching and indexing, and, 3) support random access of files within the archive.

5. **Transparency** (cf. LC SF “disclosure”, “impact of patents”, and “transparency”): any tools and specifications involved in the format should be a publicly published open standard and non-proprietary to avoid restrictions regarding activities that support long-term preservation and access of material in the archive, e.g. making modifications to the format, distributing new versions, and tracing accountability and authenticity.

6. **Flexibility of embedding metadata** (cf. LC SF “self-documentation”): the format should, if possible, support the possibility of embedding arbitrary metadata with the data objects.

7. **Flexibility in handling data** (cf. LC SF “external dependencies”): the format must be able to capture:

- data objects in its entirety or in small portions
- any media type (e.g. text, image, audio, video, executable)
- any source of material (e.g. entire disk contents, folders, files, webpages, websites) whether it is acquired through the network or provided on storage media.

Also, the stored data should be accessible using a variety of methods, environments, and operating systems.

3. Comparison of storage formats

In this section we compare several file formats that have been widely accepted as formats for storage of information, with respect to the attributes identified in Section 2. More specifically, we will broadly consider the landscape in terms of five format categories:

1. Formats for archiving content, mostly intended for aggregating, storing, transferring, and backing-up the content (e.g. tar [A26], International Internet Preservation Consortium WARC [A27], AXF [A5]).
2. Formats that capture raw data, including or excluding unused portions, as it is on the disk, mostly intended for recovery or installation (e.g. partimage [A20], dd raw image [A10]).
3. Combination of formats for archiving content or capturing raw data (e.g. those listed in item 1 and 2) with standard compression tools (e.g. gzip [A14], zip [A27], bzip2 [A7], lzma [A18]).
4. Common formats that combine archiving and compression (e.g. 7-zip [A23], SEA ARC [A4], cfs [A8], kgb [A17], PeaZip [A21]).
5. Forensic disk image formats (e.g. aff [A1], aff4 [A2]).

The examples listed above are not meant to constitute an exhaustive list of storage file formats by any means. Some formats (e.g. the EnCase image format [A12] and other proprietary formats for forensics, and rar [A22] format for archiving content) were omitted because they were clearly restricted and closed proprietary formats. Also, formats with unclear license identification (e.g. BagIt [A6]), formats which have a stable extended version (e.g. Internet Archive ARC [A3], now extended by the ISO standard WARC [A27]), and formats that are designed for limited purposes (e.g. jar [A16] for java applications and associated libraries, and iso image [A15] for optical media) have been excluded. Formats like cpio [A9] is also not extensively discussed here as it is more of a tool to access different archival formats than an archival format in itself.

Some formats have little documentation and support, because it is associated to a linux native command (e.g. shar [A25] and dd raw image [A10]), old (e.g. SEA ARC [A4]) and/or not widely adopted (e.g. cfs [A8] and kgb [A17]). While we have mentioned them in some of our discussion, the lack of documentation and support would suggest them to be unsuitable in a large scale preservation context. Likewise, formats that have no more development planned (e.g. forensic format ggzip [A13], frozen since 2006), those tied to a specific program (e.g. sgzip [A24], native format of forensic software PyFlag) or specific platform (e.g. dmg [A11] for MAC OS X) seem undesirable for serious consideration.

In view of scalability, it is not practical to consider the example formats listed in 1 and 2 without an accompanying compression method. We will therefore consider these in combination with a selected compression method (we have excluded less used compression methods such as xz-utils [A28], lzop [A19]) and group them with those examples listed in 4. The formats tar, 7-zip and PeaZip have been compared on the basis of compression size and compression speed by others⁷ who have found that,

⁷<http://warp.povusers.org/ArchiverComparison/>

6 Digital forensics formats

while 7-zip produces the best compression size, tar+bzip2 and tar+gzip show the best size to speed ratio. Other studies^{8,9} which compare gzip, bzip2 and lzma compression methods confirm that, while lzma outperforms the other two in terms of compression size, gzip is significantly superior to the other two in terms of compression and decompression speed. The gzip compression method also has the least demanding memory requirements. While there is no information on compression ratios for WARC, or AXF in combination with bzip2, gzip, and zip, as WARC and AXF are container formats that do not make special provision to optimize size of embedded objects beyond the capability of selected compression algorithm, it cannot be expected to greatly outperform tar in terms of compression size. We could not find a direct comparison of compression size and speed between the above formats and forensics file format aff, however, we do know that the compression algorithms supported within aff are zlib and lzma. The former has a typical compression ratio of 2:1 to 5:1¹⁰ which is comparable to that of gzip, and, the latter compression algorithm is shared with 7-zip. This suggests that its potential to compete with the best content archiving format. There is also the added benefit that aff provides settings to control the quality, speed, and size of output data.

In the Section 3.1, we have presented a general discussion on file formats with respect to the seven attributes that we have identified. We have followed up on the discussion in Section 3.2, with a direct comparison between tar, WARC, and aff, three of the most promising formats listed above. While AXF also claims to be an open standard conforming to preservation aims, it is a very new development, with a lack of access to detailed documentation and source code, making it difficult to assess. For this reason, we have reserved judgement on this format with regard to its suitability for inclusion in a large scale archival initiative.

3.1 General discussion of file format attributes

In this section we first present some broad observations on various formats with respect to the attributes identified in Section 2.


Completeness of data

There are different degrees of information being archived in all the different formats listed. For example, tar will save systems information such as permissions and file directory structure, others such as partimage have limitations on supported file systems, and does not retain information from unused sectors. Formats like 7-zip doesn't retain file permissions across platforms, e.g. if your data was on a Windows system and you make an archive and copy it onto a linux machine all permissions will be reset. There are many hidden issues of this nature. The inability to retain information of this sort also manifests in formats such as WARC which is designed to aggregate resources on the Internet in a descriptive surface oriented fashion without much regard to original file system structure or the file system characteristics of the embedded resource (e.g. image). In contrast, forensics formats take maximum caution to keep the data as it was at the time of creation as this can constitute vital evidence in court.

⁸<http://tukaani.org/lzma/benchmarks.html>

⁹http://blog.i-no.de/archives/2008/05/08/index.html#e2008-05-08T16_35_13.txt

¹⁰http://www.zlib.net/zlib_tech.html



Recovery and validation

Publicly available information on archive file formats (excluding WARC and AXF) show that only shar, ace, afa, arj, DGCA, WinMount format, rar, ultra compressor II come with support for integrity check, recovery record, and encryption support¹¹. These formats are proprietary, of unknown license, poorly documented (e.g. shar) or have a limited community of support (e.g. DGCA). The WARC format, as far as we know, does not have any validation or encryption mechanisms built into the format. In comparison, forensic disk images (e.g. aff) almost always come with some means of supporting all three as it forms the basis of admissibility of the extracted information as evidence in court. While the Archival eXchange Format (AXF) does provide validation mechanisms, its provisions for recovery, i.e. robustness against errors are yet to be tested. In fact, while, with many formats, the corruption of part of the data leads to the loss of a big chunk of data, formats like Advanced Forensics Format (aff) have provisions for the restoration of maximum amount of the uncorrupted data.

Scalability

Many of the listed formats have limitations on the size of the input and output file that they can produce (older versions of tar only allowed up to a file size of 8 gigabytes). The elasticity and processability of a format are key aspects of their scalability. Even some forensic file formats came with this limitation. However, unlike forensic file formats, most of the other formats do not allow easy partition of the data to be archived into blocks of manageable size. In addition, newer versions of forensic file formats such as Advanced Forensics Format (aff) have lifted the limitation on file size. More importantly, however, some archival formats (e.g. tar) do not allow random access to data, i.e. there is no way to retrieve individual files without unpacking and decompressing everything. This will incur a significant overhead for management (for example, migration of selected file types within the archived object), indexing, and retrieval operations within the archive. Even when the format allows random access (e.g. 7-zip), it is often the case that the selected file has to be decompressed before processing. Forensics formats like aff, in contrast, allows searching and analysis of the data without any decompression.

Flexibility

In terms of metadata, both WARC and AFF are designed to support arbitrary metadata. The format tar and other content archival formats, partimage and dd raw image support only limited amount of metadata. This is natural as content archival formats and raw disk images are generally born as a means of storing and transferring data from one location to another, while WARC and forensics formats are designed to support data access and analysis by end-users as well as storage and transfer.

With respect to flexibility across platforms, while many of the listed formats support multiple platforms, tar requires third party tools on Windows which may incur extra cost in terms of processing time, and pose potential obstacles for long term preservation as the third party tools are often not open source. One clear disadvantage

¹¹http://en.wikipedia.org/wiki/Comparison_of_archive_formats

8 Digital forensics formats

of aff is that it assumes the image is from a disk as opposed to a collection of files or folders. This however is not an insurmountable obstacle, as the harvest websites can be, in theory, mounted on to virtual disk which are then turned into images using aff (see Figure 1). Further, an extension of aff, known as aff4, now allows the capture of webpages over the network as images. It may be too soon for aff4 (it may not be stable enough) to be employed, but the format promises to be compatible with aff formats, i.e. a plan to initially use aff with views to migrate to aff4 when it becomes stable is fully feasible.

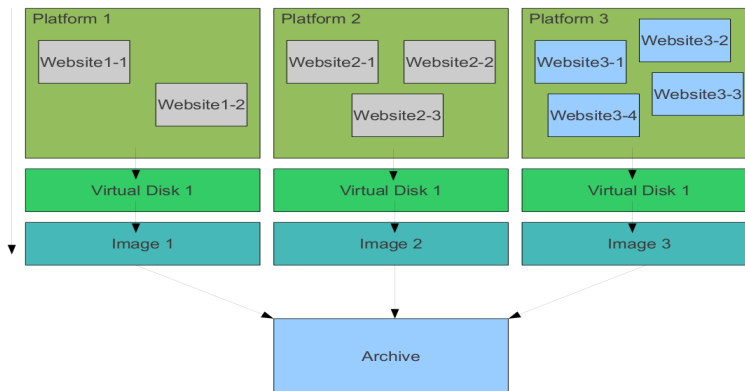


Figure 3.1.1 Workflow: implementation of aff format using virtual disks.

In addition to what was mentioned above, the International Internet Preservation Consortium WARC format has been shown to have compatibility issues with the Internet Archive ARC format even though it was created to accommodate previous data stored in the Internet Archive ARC format¹². And, data recovery problems have been observed with respect to tar¹³.

3.2 Comparison of tar, WARC, and aff


In Table 3.2.1, we have summarised aspects of the seven attributes with respect to three file formats: tar, WARC, and aff.

Table 3.2.1. Comparison of seven attributes across formats tar, WARC and aff.

Attribute	tar	WARC	aff
Completeness	partial File structure preserved but not other dependencies and change history.	no	yes
Recoverability	no	yes	yes

¹² <https://webarchive.jira.com/wiki/display/Heritrix/ARC+to+WARC+%28to+ARC%29>

¹³ <http://www.linuxquestions.org/questions/linux-software-2/recovering-files-from-corrupt-tar-archive-326716/>



In-built validation	possible with gzip	no	yes
Scalability	no Have to unpack everything before it can be searched or indexed. May have limits on size if it becomes huge.	partial No information on whether it can be searched without unpacking and decompressing.	yes
Transparency	yes	yes	yes
Flexibility embedding metadata	no	yes	yes
Flexibility handling data	partial Cannot control file sizes. Access possible using several software, but, software might be proprietary.	partial Rendered accessed only by Internet archive software. As it does not interact with embedded data, size may be difficult to control	partial Input data only in the form of disks. Easy manipulation of data chunk size. Access possible using several access software.

The description in Table 3.2.1 illustrates that:

- the tar format has limited provisions for validation or recovery mechanisms, and no support for metadata. While the format allows working with various media types and collections, it does not allow arbitrary block sizes. The format does retain file structure information and permissions, but it does not retain sector by sector information including free space.
- while WARC is specific to web crawls and therefore may provide features that are not available to other generic formats, the biggest drawback for this format is that rendered access is available only using the Internet Archive Way-Back Machine.
- the Advanced Forensic File (aff) format is clearly the most robust, in that it stores sector by sector information, as a sequence of arbitrary block size, designed for maximum recovery when error is found, has an in-built validation mechanism, and allows arbitrary metadata.

Another attractive feature of the aff format is that the collection can be searched and indexed without decompression or unpacking. While the aff format is limited to imaging disks, it has already been pointed out, in Section 3.1, this can be partially circumvented with the use of virtual disks.

4. Conclusions

In this document, we made some observations on the advantages of employing forensic file formats (more specifically, the aff format) in a digital archive. We have:

- discussed attributes for file formats that need to be considered within an archive to support digital preservation (Section 2),
- compared a broad range of file formats with respect to seven core file format

10 Digital forensics formats

attributes that we have identified (Section 3.1),

- made a direct comparison of three of the file formats, tar, WARC, and aff (Section 3.2), and,
- proposed the Advanced Forensic File (aff) format, as the most robust among the three formats as a data-mining aware preservation storage format for a web archive.

While the aff format was originally intended for use in imaging disks (Garfinkel (2006); Panda, Giordano & Kalil (2006)), we have illustrated, in Section 3.1, that this limitation can be partially overcome through the use of virtual disk technology. The virtual disk approach would not capture all the information available at the time of creation (which is often beyond our reach in the web environment), however, it helps us to work towards preserving the information we gather at the time of capture. This serves the purpose of not only supporting the preservation of the targeted information but also serves to record the process by which we have gathered and processed the information, as the data history will be preserved in the aff disk image.

In digital forensics, the fidelity, integrity, and authenticity of the data is crucial as it directly links to the admissibility of the object content as evidence in court^{14,15}. The forensics community is sensitive to the vital role that tracing data history (e.g. provenance of the data and how the data was changed plays in understanding accountability and discovering evidence). The discipline's focus on not tampering with the data, even at the time of searching (e.g. no decompression and unpacking of the storage), helps to retain the integrity of the data. As such, the handling of data within digital forensics is centred around preservation aims. Further, as forensics often involves making connections between several information entities it is rapidly opening up to supporting data mining techniques (e.g. see Louis & Engelbrecht (2011)). The possibility of processing data in an archive without unpacking and decompressing reduces overhead in implementing these processes. It is also a valuable property with respect to basic large dataset indexing and search which are must-preserve functionalities within the web data context. By absorbing digital forensics technology into the archival storage architecture, we could bring together the strengths of digital forensics that focuses on preserving digital information as evidence (data and interaction), and the wider context of preserving digital information, to introduce a preservation approach that also supports future data mining potential. The main questions to be answered to carry out the adoption of aff are: how will information be captured into virtual disks (e.g. will blogs from one website be kept together?), and how will the information within each object be segmented and distributed? We suggest that, as the first next step, a small-scale experiment be conducted to compare the formats tar, WARC and aff, (and possibly AXF format which has not been properly examined here) with respect to selected blogs harvested from the web, using compression size, speed, and preservation attributes as evaluation criteria.

Acknowledgements

The research leading to the discussion in this paper was conducted as part of the BlogForever project funded by the European Union's Seventh Framework Programme (FP7-ICT-2009-6) under grant agreement n° 269963.

¹⁴ http://www.theregister.co.uk/2011/03/01/self_destructing_flash_drives/

¹⁵ <http://www.jdfsl.org/subscriptions/JDFSL-V5N3-Bell.pdf>



References

- [book chapter] Abrams, S. (2007) “Instalment on File Formats.” in *Curation Reference Manual*, UK Digital Curation Centre.
<http://www.dcc.ac.uk/resources/curation-reference-manual/completed-chapters/file-formats>
- [conference paper] Arms, C. and Fleischhauer, C. (2003) “Digital Formats: Factors for Sustainability, Functionality, and Quality.” in: *IS& T Archiving Conference, Society for Imaging Science and Technology*, Washington, DC, 2005, 26–29 Apr.
http://memory.loc.gov/ammem/techdocs/digform/Formats_IST05_paper.pdf
- [book] Batini, C. and Scannapieco, M. (2006) *Data Quality: Concepts, Methodologies and Techniques*. Berlin: Springer.
- [report] Brown, A. (2008) *Digital Preservation Guidance Note 1: Selecting File Formats for Long-Term Preservation*. UK National Archives, DPGN-01, 2.
<http://www.nationalarchives.gov.uk/documents/selecting-file-formats.pdf>
- [report] Buckley, R. (2008) *Technology Watch Report: JPEG 2000 - a Practical Digital Preservation Standard? Digital Preservation Coalition Technology Watch Series Report*.
http://www.dpconline.org/component/docman/doc_download/87-jpeg-2000-a-practical-digital-preservation-standard
- [report] Christensen, S. S. (2004) *Archival Data Format Requirements*. The Royal Library, Copenhagen, Denmark, The State and University Library, Århus, Denmark. http://netarchive.dk/publikationer/Archival_format_requirements-2004.pdf
- [conference paper] Dappert, A. and Farquhar, A. (2009) “Significance is in the Eye of the Stakeholder.” in *European Conference on Digital Libraries (ECDL)*, Berlin Heidelberg, (September/October 2009), M. Agosti et al. (Eds.): ECDL 2009, LNCS 5714, pp. 297-308.
<http://www.bl.uk/aboutus/stratpolprog/ccare/pubs/2009/ipres2009-Dappert%20and%20Farquhar.pdf>
- [report] Fanning, B.A. (2008) *Technology Watch Report: Preserving the Data Explosion: Using PDF*. Digital Preservation Coalition Technology Watch Series Report.
http://www.dpconline.org/component/docman/doc_download/86-preserving-the-data-explosion-using-pdf
- [journal article] Garfinkel (2006) “AFF: a new format for storing hard drive images.”, *Communications of the ACM*, Volume 49, Number 2, pp. 85-87.
<http://cacm.acm.org/magazines/2006/2/6013/fulltext>

12 Digital forensics formats

- [conference paper] Guttenbrunner, M., Wieners, J., Rauber, A., and Thaller, M. (2010) “Same Same But Different – Comparing Rendering Environments for Interactive Digital Objects.” in *M. Ioannides (Ed.): EuroMed 2010, LNCS* 6436, pp. 140–152, Springer Verlag, Heidelberg.
<http://www.euromed2010.eu/e-proceedings/content/full/140.pdf>
- [conference paper] Hedstrom, M. and Lee, C.A. (2002) “Significant properties of digital objects: definitions, applications, implications.” in *Proceedings of the DLM-Forum 2002*, pp. 218-223.
http://www.ils.unc.edu/callee/sigprops_dlm2002.pdf
- [report] Huc, C. et al. (2004) *Criteria for evaluating data formats in terms of their suitability for ensuring long term information preservation*, v.5, Groupe Pérennisation des Informations Numériques (PIN).
http://www.ssd.rl.ac.uk/ccdsp2/mon04/long_term_preservation_criteria.doc
- [journal article] Louis, A.L. and Engelbrecht, A.P. (2011) “Unsupervised Discovery of Relations for Analysis of Textual Data.”, *Digital Investigations*, Volume 7, pp 154-71.
- [report] McLellan, E. P. (2006) *General Study 11 Final Report: Selecting Digital File Formats for Long-Term Preservation*. InterPARES 2 Project.
[http://www.interpares.org/display_file.cfm?doc=ip2_file_formats\(complete\).pdf](http://www.interpares.org/display_file.cfm?doc=ip2_file_formats(complete).pdf)
- [journal article] Panda, B., Giordano, J., Kalil, D. (2006) “Next-generation cyber forensics.”, *Communications of the ACM*, Volume 49, Number 2, pp. 44-47.
<http://cacm.acm.org/magazines/2006/2/5997/fulltext>
- [journal article] Pipino, L. L., Lee, Y. W. and Wang, R. Y. (2002) “Data quality assessment.”, *Communications of the ACM*, Volume 4 (2002), pp. 211–218 .
<http://web.mit.edu/tdqm/www/tdqmpub/PipinoLeeWangCACMApr02.pdf>
- [report] Rog, J. and van Wijk, C. (2008) *Evaluating File Formats for Long-Term Preservation*. Koninklijke Bibliotheek National Library of Netherlands.
http://www.kb.nl/hrd/dd/dd_links_en_publicaties/publicaties/KB_file_format_evaluation_method_27022008.pdf
- [report] Todd, M. (2009) *Technology Watch Report: File formats for preservation*. Digital Preservation Coalition Technology Watch Series Report 09-02.
http://www.dpconline.org/component/docman/doc_download/375-file-formats-for-preservation

Appendix I Table of archival file formats and compression utilities

Reference Number	Acronym	Expansion	Developers	Description	URL
A1	aff	advanced forensics format	Simson Garfinkel & Basis Technology	Extensible open format for the storage of disk images and related forensic metadata, using segments.	http://www.forensicswiki.org/wiki/AFF
A2	aff4	advanced forensic framework 4	Michael Cohen, Simson Garfinkel, & Bradley Schatz	Evidence management system integrated within the file specification	http://www.forensicswiki.org/wiki/AFF4
A3	Arc (IA)	internet archive archive format	Internet Archive (Mike Burner & Brewster Kahle)	Format of aggregate files. It must be possible to concatenate multiple archive files in a data stream.	http://www.archive.org/web/researcher/ArcFileFormat.php
A4	ARC (SEA)	system enhancement associates archive format	System Enhancement Associates (Thom Henderson)	Lossless data compression and archival format. Legacy format incapable of compressing entire directory trees.	http://www.fileformat.info/format/arc/corion.htm
A5	AXF	archive exchange format		The AXF object contain payload accompanied by structured or unstructured metadata, checksum and provenance information, full indexing structures in an encapsulated package.	http://www.openaxf.org/
A6	BagIt		California Digital Library	Storage and network transfer of arbitrary digital content, using file system directories. A "bag" consists of a "payload" (the arbitrary content) and "tags", which are metadata files intended to document the storage and transfer of the bag.	http://tools.ietf.org/html/draft-kunze-bagit-06
A7	bzip2		Julian Seward	Lossless data compression algorithm that uses the Burrows–Wheeler transform to convert frequently-recurring character sequences into strings of identical letters.	http://bzip.org/
A8	cfs	compact file set	Pismo Technic Inc.	Open archive file format and software distribution container file format. Mostly for reading optical media.	http://www.pismotechnic.com/cfs/
A9	cpio	copies (cp) into or out of (io) archive	Originally Unix, later GNU version developed	Tape archiver as part of PWB/UNIX. Later developed into GNU cpio. Usually tar is now preferred.	http://www.gnu.org/software/cpio/cpio.html
A10	dd raw image	disk duplication	Originally Unix, later made	Raw sector-by-sector image data. No metadata data. No built-in compression.	http://linux.die.net/man/1/dd

14 Digital forensics formats

			available on Linux distributions.		
A11	dmg		Apple Macintosh	MAC OS X disk imaging format.	Wikipedia article: http://en.wikipedia.org/wiki/Apple_Disk_Image
A12	EnCase image format		EnCase	Closed format used by EnCase based on ASR Data's Expert Witness Compression Format.	http://www.forensicswiki.org/wiki/EnCase
A13	gfzip		gfz project	Forensics File Format, allowing non-sequential access, development frozen since 2006.	http://gfzip.nongnu.org/filespec.html
A14	gzip	Gnu zip	GNU project (Jean-Loup Gailly & Mark Adler)	Compression algorithm based on a combination of Lempel-Ziv (LZ77) and Huffman coding.	http://www.gzip.org/
A15	iso image	ISO 9660:1988, ECMA-119		Optical media disk imaging format.	http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=17505
A16	jar	java archive		Format for aggregating java class file library.	http://download.oracle.com/javase/6/docs/techno/guides/jar/jar.html
A17	kgb	KGB Archiver	Tomasz Pawlak	Compression and archiver based on the PAQ6 algorithm.	http://kgbarchiver.net/cgi-sys/suspendedpage.cgi (found the site to be suspended at the time of writing this paper)
A18	lzma	Lempel–Ziv–Markov chain algorithm	Igor Pavlov – some question whether Pavlov is the creator.	First used in the 7z format of 7-zip. Default compression method used in 7-zip.	http://www.7-zip.org/
A19	lzop		Markus F.X.J. Oberhumer	Lossless data compression library written in ANSI C that favours speed over compression ratio.	http://www.lzop.org
A20	partimage	partition image	Francois Dupoux & Franck Ladurelle	Disk cloning utility for Linux/Unix for the purpose of recovery. Limited to supported file system types and does not clone unused portions.	http://www.partimage.org/
A21	PeaZip		PEAZIP SRL	File archiver for Windows and Linux.	http://www.peazip.org/
A22	rar	Roshal ARchive	Eugene Roshal	Proprietary compression utility with a closed algorithm. Owned by Alexander L. Roshal.	http://www.rarlab.com/
A23	7-zip		Igor Pavlov	7-zip a utility with native archiving format 7z which uses lzma compression algorithm.	http://www.7-zip.org/
A24	sgzip		Australian Department of Defence	Native forensics file format for PyFlag.	http://www.forensicswiki.org/wiki/Pyflag
A25	shar	shell	Unix	This is utility for creating a shell	http://linux.die.net/man/

		archive		script. Running the script will recreate the files. Currently tar is preferred because executables pose risk to the system. Related to GNU Sharutils.	l/shar
A26	tar	tape archive format	Originally Unix command. Later developed into GNU versions.	The format was created tape backup purposes in the early days of Unix and standardized by <i>POSIX.1-1988</i> and later <i>POSIX.1-2001</i> . Later developed into the widely distributed GNU tar.	http://www.gnu.org/software/tar/
A27	WARC	web archive format	International Internet Preservation Consortium	Next generation (taking after Internet archive's Arc format) aggregated file format.	http://archive-access.sourceforge.net/warc/
A28	xz-utils		The Tukaani Project	Free compression software including LZMA and xz for UNIX-like operating systems.	http://tukaani.org/xz/
A29	zip	Originally coined to convey "speed"	Phil Katz	Created to replace ARC by System Enhancement Associates (see above). Originally part of PKZIP for Microsoft Windows.	http://www.pkware.com/documents/casestudies/APPNOTE.TXT