Tools for Smaller Budgets:
Using Open Source and Free Tools for AV Digital Preservation Workflows

Kara Van Malssen
Senior Consultant
AVPreserve

Originally published in AV Insider, Issue 2, September 2012
https://www.prestocentre.org/library/resources/av-insider-2-preservation-times-precarity

---

Digital preservation of audiovisual material is a complex, resource intensive undertaking. The essential functions of audiovisual archives demand skilled staff, specialised hardware and software, and carefully managed workflows. In an era of constraint and austerity, it is harder and harder to argue for the necessary funds for equipment, services, and staff. For smaller archives, finding a sustainable approach to long-term digital AV preservation is especially tricky.

Archiving and preservation consists of technology, people and policies. For technology in particular, digital AV archives are largely indebted and beholden to a few sizable industries: cinema, broadcast, and information technology. Commercial interests catering to the aforementioned industries have produced a seemingly attractive toolset that has the potential to provide archives with the ability to apply their policies in service of preservation-oriented workflows. Yet, even in the hands of larger well-resourced organisations, employing these tools can be challenging and resource intensive. How can smaller, resource-constrained AV archives efficiently apply cost effective tools and technologies to their workflows?

Though at first glance all tools of the heavyweight industries mentioned above appear very expensive and proprietary, this is not necessarily the case. Often, simple, free and open source tools make up an important part of the landscape. Large hardware manufacturers, software developers, and IT experts regularly utilise various forms of open source technology. Common uses include operating systems (such as Linux in server environments) and databases (such as MySQL). Fundamental IT tasks are often performed using commands built into open source operating systems or small open source tools. When integrated into large systems or used in combination these simpler options can become powerful, automated systems. Open source technologies often form an important part of the backbone of many sophisticated tools of larger, related industries. When employed individually and applied using archival policies to preservation workflows, these simple tools can also be very useful, and often less costly than the alternatives.

There are a number of reasons why archives might be attracted to the idea of open source, such as support from a community of users rather than a commercial vendor, the promise of free downloads and licenses, or the availability of source code for modification (the definition of open source). There is, however, often a misconception of the meaning of the word "free" frequently heard along with open source. As Richard

Stallman, founder of Free Software Foundation[1] and the GNU Project[2] notes, "When we call software 'free,' we mean that it respects the users' essential freedoms: the freedom to run it, to study and change it, and to redistribute copies with or without changes. This is a matter of freedom, not price, so think of 'free speech,' not 'free beer.'"[3] Technologies that are "free" as in "free beer" but do not adhere to these principles (i.e. they do not provide access to source code, and/or do not allow modification and reuse of code) would be considered *freeware* rather than open source. Freeware tools, however, may also be of use to AV archives.

While free and open source software is often free to use, as in the developers don't charge users a fee, this doesn't always mean it doesn't require other resources to effectively implement and use. For AV archives in particular, the successful utilisation of open source software to solve day-to-day operational challenges related to preservation and access still necessitates human resources in order to successfully support archival goals.

Open Source Tools in AV Archives

Funding may be the resource most obviously limited in smaller archives, but these institutions may be rich in other resources. Skilled staff, access to partnerships, and access to training are valuable resources that can be leveraged to help an archive improve its digital preservation and access initiatives. These resources can be applied to help the audiovisual archive perform its essential functions—ingest, storage, data management, preservation planning, access, etc.—with the help of open source tools.

Open source tools are typically accessible two ways: through a graphical user interface (GUI) or via the command line (CLI). We are all familiar with GUIs, which are the desktop, mobile, or web applications we interact with every day. CLI tools are more familiar to computer programmers, systems administrators, and other IT professionals. The command line interface, sometimes known as the Terminal (Mac OS)[4], Command Prompt (Windows)[5], or shell (Linux and UNIX)[6], allows users to directly access the operating system and installed programs by typing commands, without the intermediary of a GUI. Running tools on the command line can be flexible, powerful, and allow for increased automation of tasks. This is not to say that all tasks can or should be performed on the command line; video editing, for example, as an inherently visual task, is certainly more suited to a GUI.

Maximising the use of many open source tools requires some knowledge and skills on the command line. Learning, then applying these skills, can be obtained for relatively low cost. Ultimately, armed with the technologies of the digital landscape, AV archivists will be better equipped to contribute to the discussion of digital preservation tools for audiovisual materials by contributing to community efforts to document needs of the

---

[1] http://www.fsf.org/
[2] http://www.gnu.org/
[3] Richard Stallman, "Why Open Source Misses the Point of Free Software." GNU Operating System/Free Software Foundation website, updated 2012-05-18, accessed 1 August 2012 from http://www.gnu.org/philosophy/open-source-misses-the-point.html.
[4] http://en.wikipedia.org/wiki/Terminal_(OS_X)
[5] http://en.wikipedia.org/wiki/Command_Prompt
[6] http://en.wikipedia.org/wiki/Shell_(computing)

field, develop and improve applications, and become a beta tester for new tools. Only the AV archivists can articulate their requirements; if we remain silent, how can we expect the tools to be developed for us?

Example Workflows Using Open Source Tools

The following explores a few typical scenarios for audiovisual archives, and describes how open source tools may be used to facilitate all or part of a workflow. Many of the tools described below are from the digital preservation community, and can easily be applied to AV media, and others were specifically developed for digital AV preservation. Some tools come from the broader audiovisual production and distribution communities.

For each workflow or task identified, examples of GUI and CLI tools are provided. The focus here is on tools that facilitate smaller, more specialised tasks, rather than larger systems that can perform multiple functions.

Tools to Support Activities Related to Digitisation

Digitisation is inherently an expensive process. It requires expensive hardware: playback decks for source media, analogue to digital converters, monitors and scopes, fast computers, and more. When creating archival quality digital audio and video, this equipment is mandatory.[7] If your archive has a relatively small collection, or is cash-strapped, it will not be worth investing in this equipment, and instead will be more cost effective to outsource the digitisation to an experienced vendor.

Essential archival functions surrounding digitisation, however, need not be so pricey. For instance, a number of free and open source tools exist to support the different points of a digitisation workflow where either embedding or extraction of metadata should occur.

Embedding Provenance Metadata

Documenting provenance is an important function of archives. For content that was digitised from analogue material, it can be particularly useful for end users to understand that a digital media file looks or sounds a particular way because it was digitised from an analogue source. For technicians, it is important to understand what device in the digitisation chain created an artefact in a set of digital files. For a few AV file formats, embedding provenance information directly into the file can be achieved using open source tools.

The Broadcast Wave file format (BWF), the standard format for audio preservation, provides a location within the file header to embed provenance metadata. Known as the Bext chunk, this area of the file was explicitly designed to store metadata about the originator and file creation process. BWF MetaEdit,[8] a free and open source tool developed by the U.S. Federal Agencies Digitization Working Group (FADGI) with AudioVisual Preservation Solutions (AVPS), allows archives to easily embed metadata

[7] Note that digital tape-based formats, such as DAT, Mini DV, DVCAM, and DVC Pro are already digital and do not require analogue to digital conversion. Migrating these formats to digital files can be performed by capturing directly to the computer using FireWire. The hardware costs are ultimately much lower, and more likely to be in the reach of a smaller archive to acquire in-house.
[8] http://sourceforge.net/projects/bwfmetaedit/

about the ownership, origination, and coding history of the files. Coding history, as described by the FADGI guidelines for embedding metadata, is "Designed to hold data on the digitizing process including signal chain specifics, sample rate and bit depth, and other elements […] The first line documents the analogue source recording, the second line contains data on the capture process, the third line of data records information on the storage of the file."[9]

Example tasks using BWF MetaEdit include embedding metadata into a batch of files following digitisation, examining existing embedded metadata in files, importing metadata into audio files from external sources, and outputting metadata from files as CSV or XML for use in other environments. BWF MetaEdit can be used to process single or groups of files.

A similar tool for AVI files, AVI MetaEdit[10] was recently developed by the U.S. National Archives and Records Administration and AVPS, along with a metadata schema called reVTMD,[11] for capturing and embedding coding history metadata about AVI files. Both tools are available as GUI and CLI applications for Windows and Macintosh operating systems.

Technical Metadata Extraction (i.e. Characterisation)

Knowing the properties or characteristics of your digital files facilitates collection management goals including storage planning, obsolescence monitoring, collection growth, migration, and more. By performing characterisation, or technical metadata extraction, from files after digitisation (or after acquisition of born-digital files), collection managers can create a comprehensive collection profile of their digital media. This information is also useful to provide to a digital preservation repository as part of a Submission Information Package (SIP).

There are a number of tools available from the digital preservation community that support characterisation and validation of common image and text formats, including JHOVE[12] and DROID[13]. To date, these tools have not had very good support for AV media. In the meantime, archives with AV collections often use MediaInfo[14] to perform characterisation, and more recently Exiftool,[15] as its support for AV files has been expanding. MediaInfo is available as both a GUI and CLI tool for nearly all operating systems. Exiftool is primarily a CLI tool for all operating systems, although there is a simple GUI available for Windows only.

MediaInfo reads the technical metadata embedded in media files, and presents those as text. It can read a directory of assets, but the output is text on the screen. There are

---

[9] Federal Agencies Audio-Visual Working Group, "Embedding Metadata in Digital Audio Files," version 2, approved April 23, 2012. Accessed 1 August 2012 from http://www.digitizationguidelines.gov/audio-visual/documents/Embed_Guideline_20120423.pdf.
[10] https://github.com/usnationalarchives/AVI-MetaEdit
[11] http://www.archives.gov/preservation/products/reVTMD.xsd
[12] https://bitbucket.org/jhove2/main/wiki/JHOVE2-2.0.0_Download
[13] http://www.nationalarchives.gov.uk/information-management/our-services/dc-file-profiling-tool.htm
[14] http://mediainfo.sourceforge.net/en
[15] http://www.sno.phy.queensu.ca/~phil/exiftool/

some limited options for saving and making this information usable in other applications (including PBCore, and MPEG-7 XML output for Windows GUI users), however it can provide a useful understanding of the media.

The MediaInfo and Exiftool CLI applications offer much more powerful options. Both allow XML and HTML output (and even JSON and tab-delimited output from Exiftool), which allows you to potentially import into databases, display on web pages, and script processes using the data. You can even combine this output with other open source tools for extremely useful results. As an example, try exporting metadata for a directory of files in JSON from Exiftool using the command:

```
$ exiftool -r –j [DIRECTORY] > output.json
```

Then convert the output into csv using the in2csv tool from csvkit, "a suite of utilities for converting to and working with CSV,"[16] by running:

```
$ in2csv -f json output.json > output.csv
```

Now you can sort, visualise and manipulate technical metadata about your AV collections in common applications such as Microsoft Excel, Google Docs, and more.

In addition to reading a wider range of files than MediaInfo (primarily image formats), Exiftool can both read and write metadata to files in standards including EXIF, IPTC, and XMP. It includes numerous other features,[17] such as presenting output in over fifteen languages, and outputting RDF XML, offering potential for linked technical metadata!

Tools to Support Activities Related to Ingest and Archival Storage

A fundamental task of any digital repository is to ensure the integrity and authenticity of data over time. Caretakers of digital collections are concerned with this during data transfer, which is a common time for digital files to become corrupt, and also while in long-term storage over time. This requirement is common to all digital archives, whether they manage research data, still images, or audiovisual media.

In order to monitor and manage the integrity of digital files, archives often generate checksums or hashes of the data. Checksums apply algorithms to data, and produce a unique string of characters that serve as a representation of the data in its current state. If anything about that data were to change, whether through corruption or human error, the checksum value would also change. As long as the systems creating and auditing checksums are using the same standard (such as MD5 or SHA-256), the output of a checksum function with one tool should be the same as the output in another environment using a different tool. Using these principles, checksums may be generated by the submitter, then verified by the receiver. Checksums are ideally generated and documented as early in the creation process as possible (i.e. immediately after digitisation or production).

Consistent packaging of files along with their checksums (as a Submission Information Package or Archival Information Package) according to a documented specification is a

[16] http://csvkit.readthedocs.org/en/latest/index.html
[17] http://www.sno.phy.queensu.ca/~phil/exiftool/#features

very common strategy in the digital preservation community. There are a number of open source tools available to support workflows that involve packaging of digital content along with checksums, as well as those for verification (auditing) of checksums at intervals which would be applicable to digital AV collections. A few of these are described below.

Packaging

BagIt,[18] developed by California Digital Library and the Library of Congress, is a "hierarchical file packaging format designed to support disk-based storage and network transfer of arbitrary digital content. A required tag file contains a manifest listing every file in the payload together with its corresponding checksum."[19] In other words, "Bags" (the product of the BagIt utility) are directories of files, with an inventory of the files contained, and checksums for each object in the bag.

Bags are either generated before a group of files are moved to long-term storage, to provide a consistent structure for content and metadata files, or before transfer to a repository or users. They are then verified after they are received in their new storage location. Bags are generally not created by hand; instead, creation and verification of bags according to the specification is typically done by one of a number of open source tools.

For those more comfortable with GUI applications, and/or who don't have a large number of bags to create and/or verify, the Bagger tool can be used. As defined in its documentation, "The Bagger application was created for the U.S. Library of Congress as a tool to produce a package of data files according to the BagIt specification."[20] Bagger is a cross-platform Java application and offers easy creation of bags, verification of bag contents and bag completeness, specification of checksum algorithm (i.e. MD5 vs SHA-1), retrieval of bags from a web server, updating of bags and more.

For those who prefer CLI applications, a command line interface is also available for the Bagit Library. This utility offers the same functionality as Bagger, and additional features such as the ability to split bags by file type or size. Combining BagIt with a transfer protocol such as rsync[21] allows archives to move bags (even those containing large AV files) between storage locations or to a repository in a reliable and efficient manner.

Checksum Creation and Validation

Another way to create and validate checksums for files is to use a simple checksum tool that can create and document checksums for individual or batches of files. There are a number of tools available; using these, it is fairly easy to create and/or validate checksums for your entire collection.

---

[18] Current specification (v0.97 as of July 30, 2012) is available from http://www.digitalpreservation.gov/documents/bagitspec.pdf

[19] http://en.wikipedia.org/wiki/BagIt#cite_note-ENCDEP-0

[20] From the README.txt file contained in the Bagger v2.1.2 available from http://sourceforge.net/projects/loc-xferutils/files/loc-bagger/.

[21] http://en.wikipedia.org/wiki/Rsync

For Windows GUI users, one example is Karen's Hasher,[22] which allows for the creation of checksums (output as a text file) for either individual or batches of files, using a variety of checksum algorithms. For Mac users, MD5 by Eternal Storms Software[23] is a similar tool, though it is limited to the MD5 algorithm.

Command line users can get more sophisticated with checksum creation and validation. Using md5deep and hashdeep,[24] one can compute checksums using a variety of algorithms, recursively scan entire directories, compare values and display only checksum mismatches, and create checksums at the block level (rather than entire file level). fixi,[25] another command line utility, offers similar features with the addition of being able to create and verify bags. Both tools are cross-platform. By scheduling regular audits of checksums (using the UNIX cron[26] command, for instance) with these tools, files can be checked for change or corruption behind the scenes.

Tools to Support Activities Related to Access

Providing access to digital collections should be the goal of every archive. Often, providing access means converting, or transcoding, large AV files to a smaller file size or different encoding format for ease of use and distribution.

Transcoding is a very common activity in AV Archives, for a variety of reasons. AV archives are often responsible for the digitisation and preservation of very large files, which are too unwieldy for many users (including archivists themselves) to open and playback. Additionally, archives regularly distribute digital files to a variety of platforms, each of which may require that files be delivered according to their specifications (i.e. file format, encoding format, data rate, etc.). Transcoding may be required to meet these specifications as well as create smaller files from high resolution originals. There are several useful open source and freeware tools to help support transcoding, no matter when the need for access falls into your workflow. Two such examples are described below.

Transcoding

MPEG Streamclip[27] is an excellent video transcoder and editor GUI for Windows and Mac. As described by the developer, "MPEG Streamclip lets you play and edit QuickTime, DV, AVI, MPEG-4, MPEG-1; MPEG-2 or VOB files or transport streams with MPEG, PCM, or AC3 audio (MPEG-2 playback component required); DivX (with DivX 6) and WMV (with Flip4Mac WMV Player). MPEG Streamclip can export all these formats to QuickTime, DV/DV50, AVI/DivX and MPEG-4 with high quality encoding and even uncompressed or HD video." Users can also easily create sub-clips from longer video files, export audio only or individual frames. The player allows easy viewing to support

[22] http://www.karenware.com/powertools/pthasher.asp
[23] http://www.eternalstorms.at/md5/index.html
[24] http://md5deep.sourceforge.net/
[25] https://github.com/cwilper/fixi
[26] http://en.wikipedia.org/wiki/Cron
[27] http://www.squared5.com/

clipping and transcode review. It supports single and batch processing, in case you need to convert a number of files using the same transcoding specifications. While not open source (i.e. there is no access provided to the source code), MPEG Streamclip is freeware, and as such there is no charge for its use.

A CLI option, available for nearly all operating systems is FFMPEG,[28] a powerful command line transcoder commonly found behind the scenes of larger applications due to its inclusion of the leading codec library, libavcodec. Open source applications employing FFMPEG include VLC Player, MPlayer, Handbrake, and Miro. It is also used by Google Chrome, Facebook, and YouTube.

While FFMPEG has a steeper learning curve than many of the other applications mentioned here, it offers a dizzying array of options, and precise control over the output file or files being created. FFMPEG is also responsible for the development of the FFV1 lossless codec, which is gaining support by audiovisual archives for its ability to reduce file size (from uncompressed) while maintaining mathematical reversibility (i.e. ability to fully restore the file to uncompressed) and can be opened and decoded using free and open source tools.

Voicing the Needs of AV Archives

The sample tasks and associated tools described offer just a few of the options for employing open source and free tools in audiovisual archives. When used in combination, the day-to-day workflows surrounding a digital collection can become even more simplified, automated, and effective. For instance, creating a workflow that combines FFMPEG, Exiftool and csvkit, fixi, then rsync could allow for the creation of access copies from a high-resolution original, extraction of technical metadata from master and access copies, then bag the content files along with any metadata for transfer to secure storage. By adding more tools or commands to the mix, an even wider range of options opens up.

As mentioned at the beginning of this piece, open source tools are often found under the hood of larger, more complex applications, many of which are open source themselves. CollectiveAccess,[29] for instance, is an open source web-based collection management application that supports cataloguing and management of AV collections on the backend, and public access on the frontend. CollectiveAccess utilises a number of open source utilities, including FFMPEG. Another example is Archivematica,[30] a free and open source digital preservation system that combines many of the tools described in this article and more into an integrated micro-service architecture, allowing archives a low-cost entry into preservation OAIS[31] repository development. Both of these applications support AV collections, but with the input of audiovisual archivists, could become even more suited to the community's needs.

The list below provides links to a number of open source tools, many of which will be applicable to the audiovisual community. This is just a start, so follow the links and explore. And if you can't find a solution for a particular need within these toolsets, be

[28] http://ffmpeg.org/
[29] http://www.collectiveaccess.org/
[30] https://www.archivematica.org/wiki/Main_Page
[31] http://en.wikipedia.org/wiki/Open_Archival_Information_System

sure to document your challenge on the PrestoCentre forum where you can engage with others about issues and offer insights, or comment directly on a tool's page in PrestoCentre's library. Still searching for a tool? Email editor@prestocentre.org.

Voicing the needs of audiovisual archives is crucial to the further development of tools to suit our varying needs, challenges and budgets, and should certainly be added to this list!

Tool registries:

- PrestoCentre Library Tools http://www.prestocentre.org/library/tools
- Open Planets Foundation Digital Preservation Tool Registry http://wiki.opf-labs.org/display/SPR/Digital+Preservation+Tools
- NDIIPP Partners Tool Registry http://www.digitalpreservation.gov/tools/
- CDL Micro-services https://wiki.ucop.edu/display/Curation/Microservices