**Digital Preservation Workflow: Wrapper Formats**
**Jonah Volk**
**December 2, 2009**

When establishing a digital preservation workflow, properly choosing the wrapper format to be used is among the critical decisions that must be made. Along with the codec, the wrapper determines the structure of the digital destination file, affecting, among other things, the usability of the file on certain systems and in certain software. This paper will explore some of the issues surrounding the choice of wrapper format, focusing on three digital video wrappers: QuickTime, MXF and AVI. Each format will be examined as a potential candidate for inclusion in a preservation workflow, with particular attention paid to possible risk factors, using the sustainability factors proposed by the digitalpreservation.gov website as a guide. Finally, this paper will use this analysis to offer recommendations on which wrapper formats would be best used in certain situations.

Wrapper formats, also known as container formats, specify the way in which information is stored within a digital file, generally packaging together essence tracks and metadata. Wrappers differ from codecs in that the latter specify the actual algorithm by which the data is encoded. For any given wrapper format, users generally have a number of choices of potential codecs that can be used to encode the essence contained within it; the three wrappers being examined in this paper are all capable of holding a wide range of different codecs, including both compressed and uncompressed formats.

## Sustainability Factors and Mitigating Risk

The sustainability factors, mentioned earlier, were published in an attempt to provide a decision-making structure to be used when planning for the digitization of collections at the Library of Congress. They are designed to apply to all types of media, including moving image, audio and document formats, and take into account the long-term preservation of digital information. Briefly, the factors are disclosure, the degree to which a format is open or proprietary; adoption, the extent to which the format is being used in both the preservation community and the world at large; transparency, the extent to which the format is analyzable and readable; self-documentation, relating to the metadata capabilities of the format; external dependencies, the degree to which the format requires additional software or hardware to function; impact of patents; and technical protection mechanisms.[1] This paper will largely focus on disclosure, adoption, self-documentation and external dependencies, as each of these factors has a particular relevance when applied to wrapper formats, contributing to the formats' overall profiles and suitability for usage.

When considering the ideal characteristics for a wrapper format to have, a few considerations present themselves. First, in keeping with the disclosure sustainability factor, a wrapper should ideally be open-source. Using a proprietary format can potentially cause serious problems in the long term if the company that created the format ceases to support it. However, the fact that a certain format is propriety should not necessarily be seen as a "deal-breaker." As the sustainability factors document points

[1] "Sustainability of Digital Formats."

out, disclosure also takes into account the extent to which a format, though proprietary, may be publicly documented by its developer.

A second major criterion for an ideal wrapper format is a wide adoption, not just among the general public, but within the preservation community. If a format is being used by a significant number of archival/preservation entities, or by a smaller number of significantly large archives, it suggests that the format has already been vetted for its appropriateness. Furthermore, the existence of a large user base means that pressure can be put on developers to address archival concerns and that more literature will be written on the usage of the format within an archival setting.

A third consideration would be the amount and the type of metadata that can be contained within a given format, in keeping with the self-documentation sustainability factor. Attention should be paid to the types of metadata that can be stored, with structural and descriptive metadata being especially important, to the location of the metadata within the file, and to the possibility for the metadata to be migrated out of the file, either to populate a database or as part of a larger data migration project.

Connected to this last point, the greater question of data migration should be taken into consideration, specifically whether the essence and metadata can be moved into and out of the wrapper format in question and whether this process is lossy or lossless. Obviously, the ideal format will offer the possibility of a lossless migration into and out of the wrapper, a particularly important quality for a proprietary format to ensure that the data can be rescued should the wrapper format become no longer supported at some point in the future.

## QuickTime

The first of the three wrapper formats, QuickTime, is both a wrapper format, using the extension .MOV, and a multimedia framework developed by Apple and integrated into Apple's Mac OS X operating system. It was first developed and released in 1991 and has gone through a variety of revisions in the years since, with the current version, QuickTime X, being released in 2009.[2] QuickTime has always been associated with consumer video editing software, with Adobe's Premiere, released in 1991, being one of the earliest pieces of third-party software to support the format. This association was continued when Macromedia released its Final Cut editing software in 1996 and further cemented when Apple purchased Final Cut in 1999. At the time, these consumer software products were not seen as a threat to Avid, the established professional editing software, which did not support QuickTime.[3] However, as Final Cut Pro began to expand in both the consumer and professional realms, stealing Avid's market share, the QuickTime format became more and more established as a production format, an editing format, and, increasingly, an archival format.

The current QuickTime format specification details the structures that make up a QuickTime file. The file is broken down into elements called atoms, of which there are two general types, media data atoms, which contain the actual file essence, and movie atoms, which contain the metadata about the file. Both types are subdivided into a large number of different varieties, some of which will be detailed here (the full specification can be found online at Apple's website).

---

[2] "QuickTime." Wikipedia.
[3] "How Microsoft Pushed QuickTime's Final Cut."

Each atom consisting of a header element and then the atom data, and the atoms are arranged hierarchically within a file, with separate atoms for separate video and audio elements and separate tracks. The header always consists of an integer indicating the size of the atom and a four letter string designating the atom type – for example, "moov" for a movie atom. The purpose of the movie atoms is to provide technical metadata needed for software to determine if it can play back the enclosed media data and to then assist in this playback. Thus, this information includes the codec used, the duration of the media, the frame rate, and color profile data, as well as some descriptive metadata, such as the time of the file's creation. QuickTime objects also contain a user data atom, the key storage location for user supplied descriptive metadata. The data that can be stored in this atom includes a title, the names of the writer and director, keywords, and a copyright statement.[4]

Considering QuickTime as a possible wrapper format for archival purposes, it is important to look at it in the context of the sustainability factors and ideal characteristics mentioned earlier. First, on the issue of disclosure, QuickTime is a fully proprietary format owned by Apple. While this is certainly a negative indicator, there are definitely mitigating factors in the format's favor. While QuickTime is proprietary, it has been fully documented by Apple, and the company has released developer tools to allow the community at large to work with the format to, for example, provide extended application support or alter the functionality of existing software. As a result, even if Apple ceased to provide support for QuickTime (probably an unlikely occurrence), the available

[4] "QuickTime File Format Specification."

information would allow the open-source community to create new tools to play back any orphaned QuickTime files.

Looking next at the adoption sustainability factor, QuickTime is widely used within the production and post-production communities due to the widespread usage of Final Cut Pro software, as well as, on the lower end, Apple's iMovie software. Furthermore, QuickTime's extensive user base ensures that QuickTime files can be played back in a number of different video applications, not limited to Apple's own software. However, two major negative issues are connected to the adoption factor. First, QuickTime has not necessarily caught on widely within the archival community, particularly within large-scale archives. For example, the Library of Congress only uses QuickTime for file derivatives intended for web distribution.[5] The primary reasons for this non-adoption seem to be that QuickTime is proprietary and that it does not offer the level of metadata customizability as MXF. The second negative issue relates to the proliferation of file-based cameras that are being increasingly used in the production world. While some of these, like Canon's 5D mk2 camera, utilize a QuickTime wrapper, many others, like Panasonic's popular P2 format, do not. While current workflows generally involve transcoding media to QuickTime for use in editing software, it is not inconceivable to imagine that Apple might in the near future allow for native MXF editing in QuickTime, as Adobe Premiere already does. If this were to happen, the use of QuickTime could potentially drop significantly, making it a less attractive option for archival purposes.

---

[5] "QuickTime File Format." Sustainability of Digital Formats.

Regarding metadata and self-documentation, as mentioned earlier, the format contains some technical and descriptive metadata within certain atoms. However, this metadata is somewhat limited; referring to the PBCore metadata schema, the genre, description and relation fields are not included. Furthermore, the QuickTime format cannot be modified to include other metadata fields that might be useful in a preservation context – information about the original media object, for example. On the plus side, as the metadata is located in clearly defined locations within the file, it can easily be migrated out of the file. Likewise, tools exist to losslessly migrate essence data out of a QuickTime file, should the need arise as part of a potential future migration process.

## MXF

The MXF (Material eXchange Format) wrapper format was launched in 2004, having been developed by a coalition of several partners, including the Pro-MPEG group and the AAF Association and standardized by SMPTE. It was designed for maximum interoperability between systems and compression formats, and to have a great deal of flexibility regarding metadata. It is an open, non-proprietary format based on the AAF data model, an earlier attempt to develop an interchangeable multimedia standard.[6]

The structure of an MXF file is fairly straightforward: the essence is packaged with header and footer blocks which define the file type, size and structure. Additionally, the header contains the file metadata in two parts – first, a tightly-defined structural metadata component, which ensures the file's interoperability according to the AAF

[6] Wilkerson and Devlin.

model, and second, a descriptive metadata component which is fully customizable according to the user's needs.[7]

Complicating the issue is the fact that MXF was designed to have multiple instantiations, called operational patterns, in order to offer a greater deal of flexibility to users. The two primary operational patterns are designated OP-Atom and OP1a, though higher-level patterns also exist (OP1b, OP1c, etc.). OP-Atom is the simpler of the two, specifying that each MXF file can contain only a single track of essence. Therefore, in systems implementing this pattern, video and audio data must be stored as separate files, with metadata used to link them. In the OP1a pattern, files can contain multiple tracks, allowing video and audio tracks to be stored together.[8]

MXF is, as mentioned earlier, a fully open-source non-proprietary format, thus there are no disclosure-related risks associated with it. Regarding adoption, unlike QuickTime, MXF has not achieved a significant level of penetration into the production and post-production worlds, though this is beginning to change. Panasonic's P2 file-based recording system creates MXF-wrapped files in the OP-Atom operation pattern, and both Avid and Adobe Premiere can work with MXF natively. One concern is that, unlike QuickTime, which is supported by a wide variety of media players, including Apple's own QuickTime Player, only a small number of applications exist to play MXF media, a fact which could hamper the format's widespread acceptance.

MXF is being increasingly used in both the archival and television production and broadcast worlds, with the metadata support and interoperability being cited as key

---

[7] Wilkerson and Devlin.
Devlin, "MXF."
[8] Santos, 4.

factors.[9] Specifically, MXF has been used in certain large institutions, most notably the Library of Congress, as well as other archives making use of SAMMA equipment, which is designed to create MXF-wrapped JPEG2000 files as a primary preservation format. Several large non-US archival projects are either using or have proposed to use MXF, including Frace2-News, Ciris, a Dutch television production company, and the European EDCine project.[10] As another example, the Audiovisual Archive Network (AVAN), currently in development, has announced that it will create an archival MXF specification which will be used to store all of its media.[11]

Regarding metadata storage, MXF files are highly flexible and can be designed to hold as much or as little descriptive metadata as needed. As mentioned earlier, this adaptability has contributed to MXF's growing adoption in the archival and broadcast communities. Similarly, technical metadata is included in the header and encoded with using KLV (Key-Length-Value), a highly interoperable system, allowing it to be easily migrated out of a file.[12] Finally, when looking ahead to a possible future migration out of MXF, the format's interoperability ensures that the essence can also be losslessly migrated out of the file.

## AVI

The third wrapper format to be examined, AVI (Audio Video Interleave), was introduced by Microsoft in 1992 as an offshoot of the RIFF (Resource Interchange File

---

[9] Ive.
[10] Nowak and Foessel.
Golson et al.
[11] "AVAN's Solutions."
[12] "Material Exchange Format." Sustainability of Digital Formats.

Format) meta-format that also spawned the WAV audio format. As a Windows proprietary format, AVI has frequently been associated with the Windows operating system. Unlike QuickTime and MXF, AVI has no significant presence in either the professional production or archival worlds, primarily being used for amateur production and in the file sharing community, often paired with lossy MPEG-4 codecs such DivX and Xvid. Like QuickTime, although MXF is a proprietary format, it is fully-documented and thus presents fewer potential problems related to the disclosure sustainability factor.[13]

The data in an AVI file is divided into chunks which are designated with four-character codes. The standard formatting consists of two LIST chunks for each track of media information; the first, the HDRL header chunk, defines the format of the data, while the second, the MOVI chunk, contains the essence itself.[14] The file can also contain an INFO chunk that contains descriptive metadata about the file. This INFO chunk only allows for a limited set of metadata categories which do not include (again referring to PBCore) description, publisher or relation fields. These categories do include a variety of technical metadata mixed in with the descriptive metadata, although most of these categories refer to still images, a reminder that the AVI INFO chunk is borrowed directly from its parent format RIFF.[15] For AVI to be taken seriously as a preservation wrapper, developers would probably have to create a variation on the format which would include a new chunk containing relevant descriptive metadata, just as the BWF format added the BEXT chunk to WAV.

---

[13] AVI File Format." Sustainability of Digital Formats.
[14] "AVI Riff File Reference."
[15] "Embedding Metadata in Digital Audio Files."

## Conclusions

In conclusion, based on the information presented above, it seems that MXF is the best possible wrapper format to use in an archival situation, though QuickTime should be considered as well. The key consideration leading to this decision is MXF's superior metadata capabilities. Its fully open, non-proprietary nature is also highly important, though QuickTime and AVI are both fully-documented and present few disclosure problems. The primary reason to choose QuickTime over AVI would be its broader adoption and wider base of software support. However, due to reasons listed above, it is not hard to believe that MXF could reach this level of support in the near future. In fact, as more archives, particularly large well-funded archives, adopt MXF, the chances are increased for more MXF archival tools to be developed.

**Bibliography**


"AVAN's Solutions." Audiovisual Archive Network. <http://www.archivenetwork.org/services>

"AVI File Format." Sustainability of Digital Formats, Library of Congress. <http://www.digitalpreservation.gov/formats/fdd/fdd000059.shtml>

"AVI Riff File Reference." Microsoft Developer Network. <http://msdn.microsoft.com/en-us/library/ms779636%28VS.85%29.aspx>

"A Closer Look at MXF." Avid. <http://.avid.com%2Fresources%2Fwhitepapers%2FavidMXFWP3.pdf>

Devlin, Bruce. "Metadata-Driven Workflows with MXF: A Proposal for Tagging Languages." *SMPTE Motion Imaging Journal* 117:5 (July-Aug. 2008): 38-41.

Devlin, Bruce. "MXF: The Material eXchange Format." *EBU Technical Review*: July 2002.

Edge, Bob. "File Formats for Television: The Migration Dilemma." *SMPTE Motion Imaging Journal* 113:7/8 (July-Aug. 2004): 219-24.

"Embedding Metadata in Digital Audio Files." Federal Agencies Audio-Visual Working Group. Sept. 15, 2009. <www.digitizationguidelines.gov/audio-visual/documents/Embed_Intro_090915.pdf >

Golson, Chris, et al. "MXF Practice and Reconfigurable Application Specific Computing." *SMPTE Motion Imaging Journal* 115:7/8 (July-Aug. 2006): 248-54.

"How Microsoft Pushed QuickTime's Final Cut." RoughlyDrafted.com. <http://www.roughlydrafted.com/RD/RDM.Tech.Q1.07/8AA115DC-2398-456E-9319-FE5842A41BD1.html>

Ive, John G. S. "The Material eXchange Format (MXF) and the Workflow Revolution." *SMPTE Motion Imaging Journal* 113:7/8 (July-Aug. 2004): 252-6.

"Material Exchange Format." Sustainability of Digital Formats, Library of Congress. <http://www.digitalpreservation.gov/formats/fdd/fdd000013.shtml>

Nowak, Arne and Siegfried Foessel. "Digital Film Archival Using JPEG 2000 and MXF – Formats and Practical Applications." *SMPTE Motion Imaging Journal* 118:3 (Apr. 2009): 50-6.

"QuickTime." Wikipedia. <http://en.wikipedia.org/wiki/Quicktime>

"QuickTime File Format." Sustainability of Digital Formats, Library of Congress. <http://www.digitalpreservation.gov/formats/fdd/fdd000052.shtml>

"QuickTime File Format Specification." Mac OS X Reference Library. <http://developer.apple.com/mac/library/documentation/QuickTime/QTFF/QTFFPreface/qtffPreface.html>

Santos, Ernesto. "Operational Patterns… the MXF Flavors?" *NAB2007 Broadcast Engineering Conference*. Las Vegas, NV.

"Sustainability of Digital Formats." Library of Congress. <http://www.digitalpreservation.gov/formats/sustain/sustain.shtml>

Roth, Todd. "MXF Interchange: A Progress Report." *Broadcast Engineering* 47:10 (Oct. 2005): 62.

Wilkinson, Jim and Bruce Devlin. "The Material Exchange Format (MXF) and its Application." *SMPTE Journal* 111:9 (2002): 378-84.