What is the Standard Format for Digitized Audio?

Approaches for Storing Complex Audio Objects

Nick Krabbenhoeft
New York Public Library
United States of America
nickkrabbenhoeft@nypl.org

Abstract – The best practices for representing analog audio with digital bitstreams are relatively clear. Sample the signal with 24 bits of resolution at 96KHz. The standards for storing the data are less clear, especially for media with complex configurations of faces, regions, and streams. Whether accomplished through metadata and/or file format, the strategy chosen to represent the complexity of the original media has long-term preservation implications. Best practice guides rarely document these edge cases and informal discussions with practitioners have revealed a wide range of practices. This paper aims to outline the specific challenges of representing complex audio objects after digitization and potential approaches that can be adopted by the community.

Keywords – Audio, Digitization, Object Modeling
Conference Topics – Collaboration: a Necessity, an Opportunity or a Luxury?; Building Capacity, Capability and Community

I. Introduction

The deteriorating sustainability of magnetic media has prompted many organizations to pursue digitization as their preservation strategy for audio and video collections. For example, the New York Public Library is digitizing roughly 250,000 items in order to maintain the accessibility of their contents past the deterioration of the original media and/or playback equipment.

Digitization projects generally share similar specifications for the resolution of the digitization target. Digital bitstreams should represent the original signals with the highest fidelity or at least at a greater fidelity than human senses can perceive. In the case of audio, human ears cannot distinguish frequencies higher than 20 kHz, and analog audio is generally sampled at a frequency to represent recorded frequencies up to 48kHz.

Best practice documents are less clear on how to store the bitstreams. General recommendations for keeping audio signals as uncompressed PCM streams wrapped in a Wave or Broadcast Wave format leave room for interpretation. Should left and right stereo tracks be stored in separate files or interleaved as channels in a single file? If a stream exceed the 4 GB size limit[1] imposed by the Wave header, should the stream be split into two files or stored in a single RF64 Wave file?

Reviews of the audio digitization literature have shown relatively little guidance on questions like this, and informal conversations have revealed a range of approaches. IASA TC-04 devotes three paragraphs in total to target formats. [1] The Sound Directions project documented a starting point, but only for two institutions with individual contexts. [2]

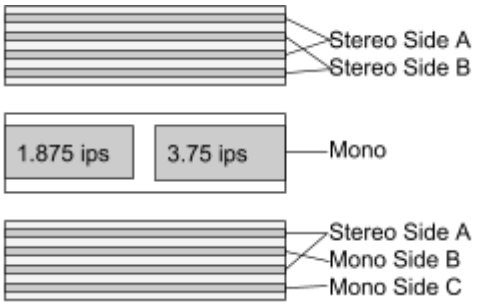
According to the OAIS Framework, organizations are responsible for defining the specifications for SIPs and AIPs, including the Content Objects those packages contain. Whether an organization is receiving files from from a digitization vendor or producing them internally, using shared approaches to the

[1] The Wave file format based on the Resource Interchange File Format (RIFF), which allocates bytes 4-7 to specifying the file size. This limits the size to 2^32 bytes (about 4.295 GB). RF64 extension defined in EBU 3306 [7], allows for daisy chaining of additional audio data in 18 EB chunks.

composition of Content Objects increases the long-term accessibility of those objects. This paper documents potential options in hopes of spurring more public discussion of these issues.

II. The Challenge of Complex Audio Objects

Magnetic media is composed of metallic particles attached to a flexible tape by a binder. Nothing inherent to the construction restricts how audio is stored. For instance, on a Compact Cassette, the particles are magnetically aligned along four track to hold the left and right channels of side A and B. Given the appropriate equipment, a Compact Cassette could also store a single, wide track of mono audio recorded at multiple speeds, or Side A of stereo and 2 additional mono tracks.



Three example layout of audio on an ⅛ inch tape.
1. 2 faces with 2 streams each
2. 1 face with 2 regions at different speed
3. 3 faces, one with 2 streams, two with 1 stream

Audio layouts created by different recording and playback scenarios have been abstracted with the following terminology. [3]

Stream - a single linear sequence of audio signals

Region - a group of streams to be played back synchronously

Face - a group of regions to be played back sequentially

These three abstractions introduce a dimensionality problem. Where still imaging programs must take at least two images to capture the front and back of an object, audio digitization programs must handle hierarchies of streams in regions in faces with no strict limit on the number of any component.

One final complication is the relationships of streams to one another. Streams may be intended to be played back individually (mono), in tandem (stereo), or across an array of speakers (surround sound).

If the goals of audio digitization programs is to preserve the information from the original media, these complexities are significant properties that should be stored as part of the Content Information.

III. Strategies for Representing Complex Audio Objects

Much like the original creators had freedom in recording audio to magnetic media, so too do collecting organizations have freedom in how to representing media as digital files. To simplify this discussion, strategies will progress through managing streams, regions, and then faces.

Streams

Perhaps the most common strategy is storing every stream as its own Wave file. For example, a 24-track open-reel master recording would result in 24 mono wave files. The relationship between these files is managed by either filename conventions, structural metadata, or both.

It bears repeating as documented in Sound Directions, “filenames are not a reliable means of storing information.” Unfortunately, there is no standard method to record this type of semantic metadata. Harvard and Indiana used AES-57 and METS respectively. [2] PBCore offers another possibility, [4] and NYPL maintains a custom schema. [5]

However, the most common multi-stream occurrence, stereo, is supported by nearly all audio formats. Many programs opt to use this feature to store stereo audio in a single file.

Broader multi-stream support is also available, but to varying degrees. In 1999, Microsoft released a specification for encoding up to 18 surround sound playback locations for multi-stream Waves. [6] The European Broadcast Union (EBU) released a specification for both increasing the 4 GB file-size limit in Wave to 16 EB (RF64) and storing 18 number of channels in Wave (MWBF). [7] As extensions of the original Wave format defined in 1992, none of the additions have universal support and in some cases are more theoretical than practical.

Other formats such as MXF and Matroska are possibilities for multi-stream containers. [1] Because of their broad usage in media industries where multiple language audio

tracks are common, playback support is more common. Streams can be grouped into mono, stereo, or surround sound configurations, and switching between a group during playback matches the timecode between the group.

Regions

Similar to streams, different regions are often stored to separate files with filenames or metadata preserving the relationship between regions.

Wave cannot store sequential audio sequences. However, Broadcast Wave includes a TimeReference field that can be used to record the temporal relationship between two files by recording their start times on a shared timeline. [8]

Sequential storage is possible in container formats such as MXF and Matroska through chapter features. Unlike Broadcast Wave this provides the ability to store multiple regions in a single file, but it lacks the metadata standard encode the temporal relationship between files. During digitization, engineers will often start before beginning of a region and stop past the end of a region. Without a timecode, it is difficult to reassemble regions onto a shared timeline

Faces

Faces generally have a sequential relationship to one another, so the same storage strategies apply. When using chapters within a container format, it might be necessary to use sub-chapters in order to represent a hierarchy of faces and regions.

IV. Discussion

There is a garden of forking paths when it comes to storing digitized audio. Specifications for digitization targets should go past 96 kHz at 24 bits per sample in a BWF, but examples of such specifications are difficult to find in best practice literature.

Greater discussion and documentation of the approaches above would be particularly useful for two communities, digitization labs and repository developers.

In the first instance, the support for custom metadata formats, embedded metadata, Wave extensions, and container formats varies across digitization software and vendors. If every collecting institution chooses its own combination of strategies, labs are forced to support that full range of strategies, increasing expense and likelihood of confusion or errors. After digitizing materials through in-house and vendor workflows, complex audio configurations is still a difficult class of media to design QC processes for. Documentation of even a few shared strategies would greatly simplify target selection for collecting organizations and support for labs.

In the second instance, representing the semantic relationship between files is one of the most challenging aspects of repository development. Documenting edge cases and migrating from previous strategies occupy outsized portions of time. Again, complex audio has presented a particular challenge for the development of ingest workflows at NYPL and, based on conversations, at other institutions as well.

While all of the summarized strategies are viable, it is from this perspective that the author find container formats to be most worth investigation. NYPL has experimented with using the Matroska format to store 24 tracks of mono audio in a single file with an image of the track-listing. Doing so proved to be far simpler for object modeling than storing the relational metadata in a sidecar and developing a parser. However, as an experiment, it bears examination if such strategies impede access in the future.

V. Conclusion

This paper is a provocation to discuss and document how digitization projects encode and package outputs. It does not believe there is a single optimal strategy but hopes that preferred strategies may be developed.

REFERENCES

[1] TC-04 Guidelines on the Production and Preservation of Digital Audio Objects (web edition). IASA, https://www.iasa-web.org/tc04/key-digital-principles.

[2] Sound Directions. Indiana University and Harvard University. http://www.dlib.indiana.edu/projects/sounddirections/.

[3] AES 57 AES standard for audio metadata - Audio object structures for preservation and restoration. Audio Engineering Society. http://www.aes.org/publications/standards/search.cfm?docID=84

[4] PBCore Part Type, PBCore. https://pbcore.org/elements/pbcorepart

[5] AMI Metadata Analog Reel Sample, NYPL. https://github.com/NYPL/ami-metadata/blob/master/versions/2.0/sample/sample_digitized_audioreelanalog.json#L48

[6] Multiple Channel Audio and Wave Files, Microsoft. http://www.microsoft.com/whdc/device/audio/multichaud.mspx
[7] EBU 3306 MBWF/RF64, European Broadcasting Union. https://tech.ebu.ch/publications/tech3306
[8] A Primer on the Use of TimeReference, AVP. https://www.avpreserve.com/wp-content/uploads/2017/07/AVPS_TimeReference_Primer.pdf
[9] https://tech.ebu.ch/publications/tech3306