Interview with Brendan Coates





]

Hey Brendan! Introduce yourself please.

Hey everybody, I'm Brendan and at my day job I'm the AudioVisual Digitization Technician at the University of California, Santa Barbara. I run three labs here in the Performing Arts Department of Special Research Collections where we basically take care of all the AV migration, preservation, and access requests for the department and occasionally the wider library. I'm a UMSI alum, I got a music production degree there too, so working with AV materials in a library setting is really what I'm all about.

And, I get to work on lots of cool stuff here too. We're probably most famous for our cylinder program, we have the largest collection of ?wax? cylinders outside of the Library of Congress at roughly 17,000 items, some 12,000 of which you can listen to online. I'm particularly fond of all the Cuban recordings from the Lynn Andersen Collection that we recently put up. We're also doing a pilot project with the Packard Humanities Institute to digitize all of our disc holdings, almost half a million commercial recordings on 78rpm, over the next 5 years.

And we're building out our video program, too. We can do most of the major cartridge formats. We're only doing 1:1 digitization though, a lot of my work these days is figuring out how to speed up the back-end - we have 5000 or so videotapes at the moment but I know that number is only going to go up.

Outside of work, Morgan Morel (of BAVC) and I have a thing called Future Days where we're trying to expand our skills while working with smaller institutions. Last year we made a neat tool called QCT-Parse, which runs through a QCTools Report and tells you, for example, how many frames have a luma bit-value above 235 or below 16 (for 8-bit video), outside of the broadcast range. You can make your own rules, too. We had envisioned it as like MediaConch for your QCTools reports and sorta got there? we're both excited to be involved with SignalServer, though, which will actually get there (and much, much further).

Today, though, I'm going to be talking about work I did with one of our clients, revising their automated ingest workflow.

What does your media ingest process look like? Does your media ingest process include any tests (manual or automated) on the incoming content? If so, what are the goals of those tests?

Videos come in as raw captures off an XDCAM, each individual video is almost 10mins long, they're concatenated into 30min segments, the segments are linked to an accession/ interview number. They chose this route to maintain consistency with their tape-based workflow. This organization has been active since the 90's, so they're digitizing and bringing in new material simultaneously, I was only working on the new stuff, but it made it organizationally easier for them to keep that consistency.

After raw captures are concatenated, we make flv, mpeg, and mp4 derivatives and they're hashed and sent to a combination of spinning discs and LTO, all of their info lives in a PBCore FileMaker database. Derivatives are then sent out to teams of transcribers/ indexers/ editors to make features and send to their partners.

When I started this project, there was no in-house conformance checking to speak of. Their previous automated workflow used Java to control Compressor for the transcodes and, whatever else might be said about that setup, they were satisfied with the consistency of the results.

Looking back on it now, I ~should~ have used MediaConch right at the start to generate format policies from that process and then evaluated my new scripts/ outputs against them, sort of a ?test driven development? approach.

Where do you use MediaConch? Do you use MediaConch primarily for file validation, for local policy checking, for in-house quality control, for quality testing for vendor files?

We use MediaConch in two places: first, on the raw XDCAM captures to make sure that they're appropriate inputs to the ingest

script (the ol' ?garbage in, garbage out?); and second, on the outputs, just to make sure the script is functioning correctly. Anything that doesn't pass gets the dreaded human intervention.

At what point in the archival process do you use MediaConch?

Pre-ingest, we don't want to ingest stuff that wasn't made correctly, which you'll find out more about later.

I think that this area is one where the MediaConch/ MediaInfo/ QCTools/ SignalServer apparatus can help AV people in archives to contextualize their work and make it more visible. These tools really shine a light on our practice and, where possible, we should use them to advocate for resources. Lots of people either think that a video comes off the tape and is done or that it's only through some kind of incantation and luck that the miracle of digitization is achieved.

Which, you know, tape is magical, computers are basically rocks that think and that is kind of a miracle. But, to the extent that we can open the black box of our work, we should be doing that. We need to set those expectations for others that a lot of stuff has to happen to a video file before it's ready for its forever home, similar to regular archival processing, and that that work needs support. We're not just trying to get some software running, we're implementing policy.

Do you use MediaConch for MKV/FFV1/LPCM video files, for other video files, for non-video files, or something else?

Each filetype has its own policy, 5 in total. The XDCAM and preservation masters are both mpeg2video, 1080i, dual-mono raw pcm audio, NDF timecode. Each derivative has its own policy as well, derived from files from the previous generation of processing.

Why do you think file validation is important?

Because it'll save you a lot of heartache.

So, rather than start this project with MediaConch, I just ran ffprobe on some files from the older generation of processing and used that to make the ffmpeg strings for the new files. As a team, we then reviewed the test outputs manually and moved ahead.

The problems with that are 1) ffprobe doesn't tell you as much as MediaConch/ MediaInfo does (they tell you crucially different stuff), and 2) manual testing only works if you know what to look for, which, because we were implementing something new, we didn't know what to look for.

It turns out that the ffmpeg concat demuxer messes with the [language] Default and Alternate Group tags of audio streams. Those tags control the default behavior of decoders and how they handle the various audio streams in a file, showing/ hiding them or allowing users to choose between them.

What that bug did in practice was hide the second mono audio stream from my client's NLE. For a while, nobody thought anything of it (I didn't even have a version of their NLE that I could test on), so we processed files incorrectly for like three months. The streams were still the preservation masters (WHEW) but, at best, they could only be listened to individually in VLC. If you want to know more about that issue you can check out my bug report.

If we had used MediaConch from the beginning, we would have caught it right away. Instead, a years worth of videos had to be re-done, over 500 hours of raw footage in total, over 4000 individual files.

It's important to verify that the things that you think you have are the things that you actually have. If you don't build in ways to check that throughout your process, it will get messy and it's extremely costly and time consuming to fix.

Anything else you'd like to add?

I'm really digging this series and learning how other organizations are grappling with this stuff. It's a rich time to be working on the QC side of things and I'm just excited to see what new tools and skills and people get involved with it.