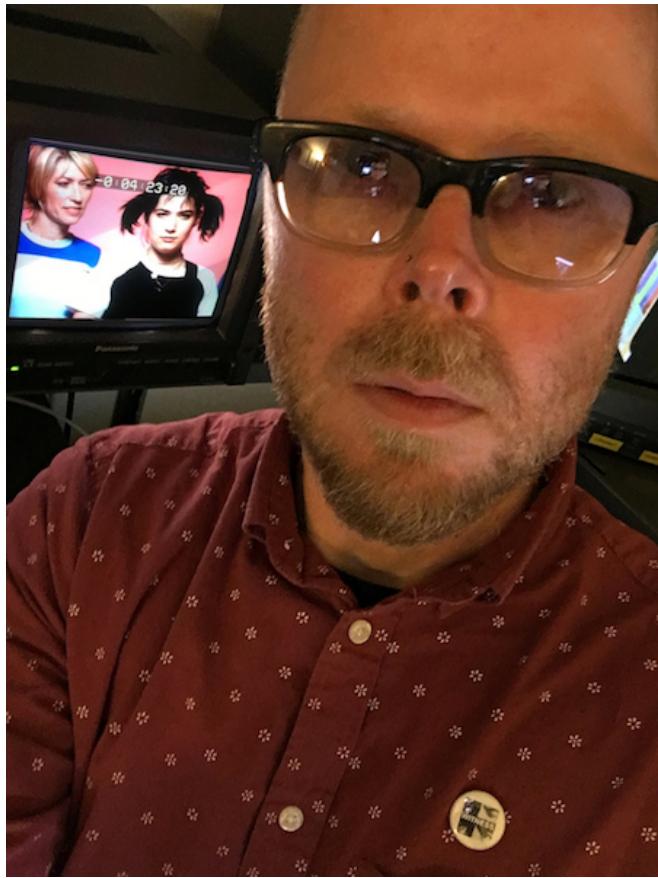
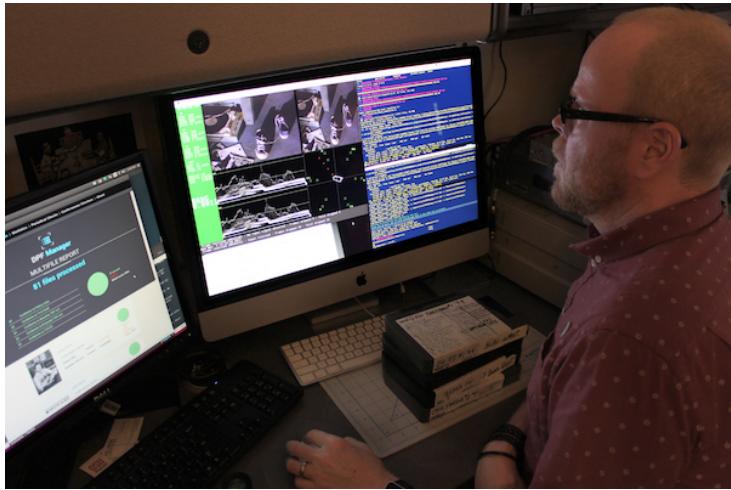


Interview with Brian E. Davis of Oregon State University



Hey Brian! Introduce yourself please.

Hi Ashley! I'm [Brian](#), the head of the [Digital Production Unit](#) for the [Special Collections & Archives Research Center](#) at Oregon State University Libraries & Press. I've been at OSU since the summer of 2012.

My background is pretty broad and a bit random. I was the visual materials archivist at Arizona State University from 2005-08 and I

worked as a digital production developer at Duke University right after leaving ASU. Through a series of unfortunate events, I spent a few years as a media services librarian at a small liberal arts college in North Carolina. Although outside the library/archives world, my earliest somewhat related work was installing exhibits at an art museum and working as movie theater projectionist. My experiences from both of those jobs still influence the work I do today. Oh yeah, I went to grad school and should mention that. I have master's in computer science and I've been involved with programming to some degree since the late 90's.

As you might expect, the Digital Production Unit provides digitization services for the library. I sometimes joke that I'm a unit of one since I'm the only staff member doing standards-based digital production work in the library. My work spans the entirety of our digital production process, from prepping physical materials and performing minor stabilization/repairs all the way through to digitization and later stage digital preservation. I'm responsible for a lot of things. Not having ALL the resources is both a good thing and a bad thing. I won't mention the bad parts of it, but the best thing about it is that I've been able to come up ways to be more efficient.

Luckily, I have three part-time student staff and between the four of us, we produce a substantial number of files for both of our repositories ([ScholarsArchive@OSU](#) & [Oregon Digital](#)), as well as for our preservation storage. In terms of day-to-day work, my students take care of most of the flatbed digitization while I tackle the materials prep, book scanning, videotape transfers, quality control, and digital preservation.



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?

I should mention up front that we don't typically work with files coming in from other sources. Virtually everything we work with is produced in my unit and we produce a large number of PDF, TIFF, and MKV files each week. Every file gets reviewed.

#NoFileLeftBehind

Neither of our digital repositories is currently being used as preservation repositories. That simply means that I make reduced-quality access files for them. That also means that I'm juggling hundreds of derivative files spread across a number of workstations and production servers, along with all the master files. It can get a bit confusing. My focus is on the preservation-level files and those are moved onto two separate ZFS storage systems via a [BagIt](#) protocol. One is a local machine that I configured primarily for video storage and the other is a network share managed by our IT department. Because these files are preservation master files, my testing is to verify that the files were produced in accordance to our local policies and to perform format-specific identification, validation, characterization, and to run fixity checks.

I don't utilize watched folders as much as I probably could, so most of the processes are somewhat manual. I prefer using

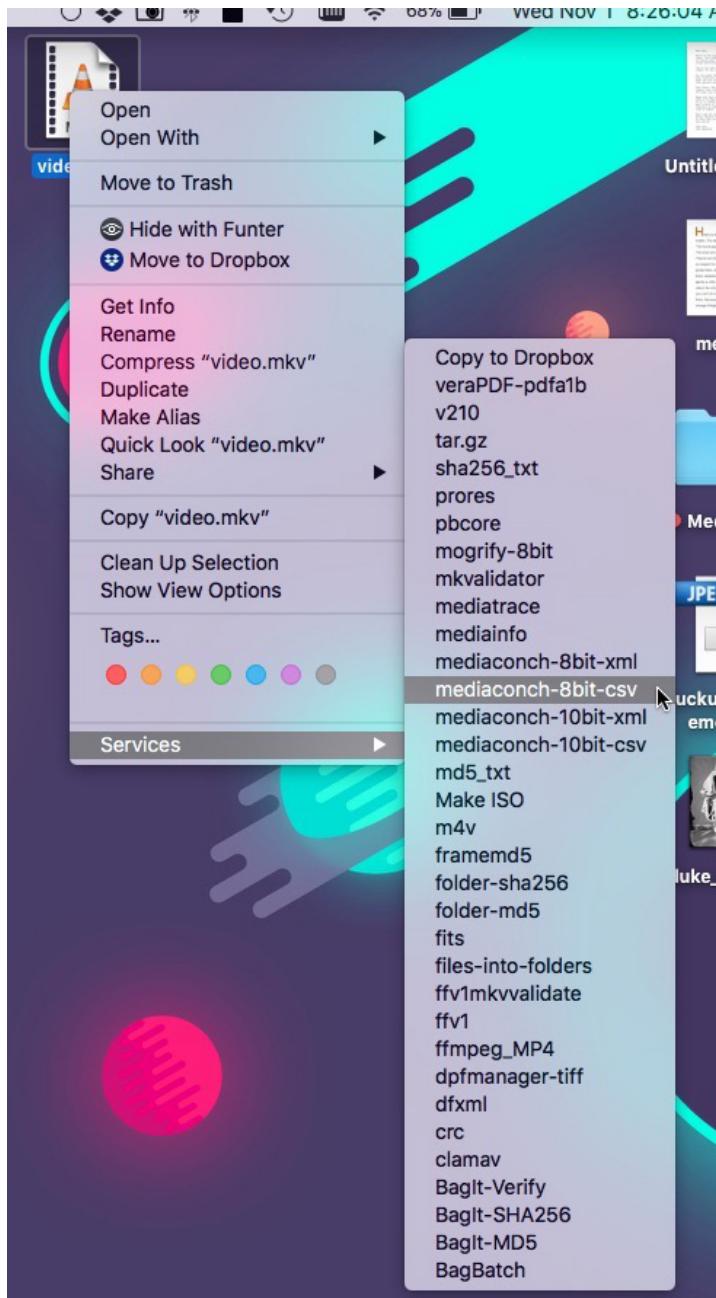
command-line tools for most things. However, I often wrap terminal commands inside AppleScripts and install them as ?right-click actions? across our macOS workstations. This is how I've implemented BagIt, [FFmpeg](#), and a few other tools we use daily. I chose this route both to ease my students into taking on some digital preservation work and also to make certain tasks easier for myself. These actions are connected to a number of command-line tools and keeping things updated is key to making it all work. Thank God for [Homebrew](#).

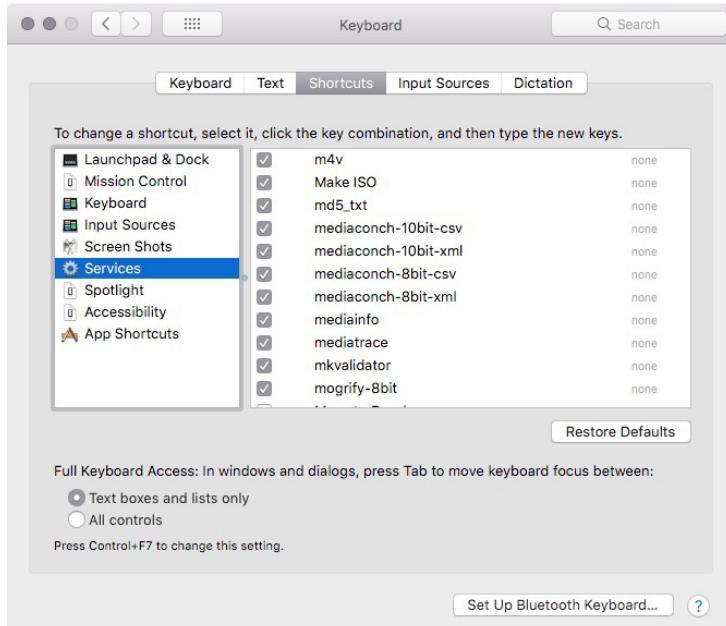
Quality control for PDF files can be burdensome and being the only reviewer, I need to find ways to be more efficient. In that spirit, I've abandoned Adobe Acrobat Pro? s Preflight and moved to using [Quick Look](#) to scroll through the PDF and validating with [MediaConch](#) (GUI). Things move quite a bit faster this way. I've used command-line versions of [veraPDF](#) and [Ghostscript](#) in the past, but I like the speed of my new workflow.

We produce hundreds of TIFF files each week - more than any other filetype. I use [DPF Manager](#) for quality control on TIFF files since it can parse file directories and determine if files are valid and whether digitization specifications have been followed based. Verifying this information in a batch process eliminates the extra steps of manually checking with Photoshop or Bridge. I generally use the command-line version of DPF Manager on my Linux machine for this task since my other two workstations are often busy processing video files and PDFs.

As someone juggling many many processes and workflows, I try to keep things as simple as I can when it comes to my videotape digitization. I use [vrecord](#) for all of my analog video transfers and capture as MKV/FFV1/LPCM. I use a variety of other tools for digital videotape like [VLC](#), [QuickTime](#), and [Final Cut Pro X](#), but I maintain format appropriate MKV/FFV1/LPCM specs for the preservation-level files. MediaConch is used to validate and to check that my transfer policies have been followed.

As soon as our files check out, I right-click to bag them and manually push them up to preservation storage.





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?

I've used MediaConch during the quality control process for my video files for a while now. I use it for validation and policy checking. I just started using it for PDF validation after struggling to find an efficient process.

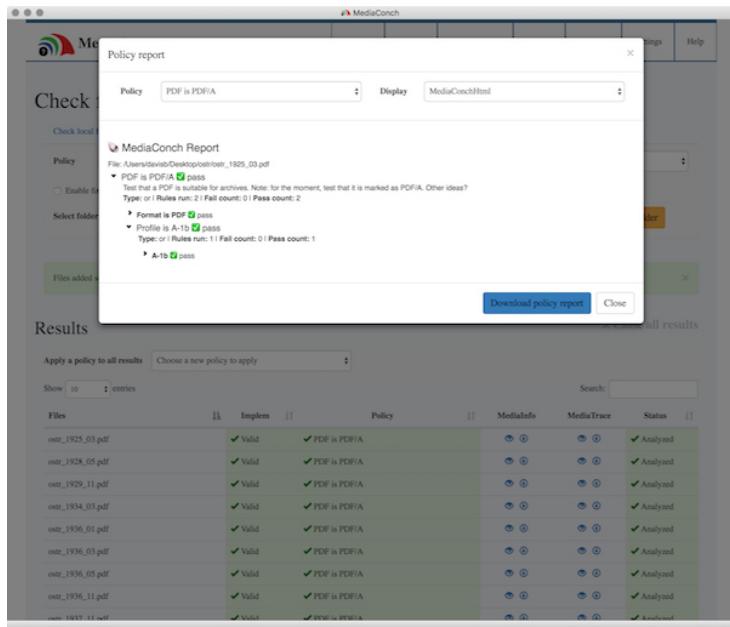
Files coming into the library from vendors do not undergo any sort of quality control process or validation check. This is because the files simply do not come through my unit. Hopefully that will change because I kind of know what I'm doing with this stuff. ^_?_/-

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

Pre-ingest. MediaConch is the first tool that I run during quality control. It's an integral first step as it determines what happens next. If the file doesn't pass, I need to figure out why and correct that problem. If it passes, then we can happily move onto the next step.

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

I use MediaConch for all of my MKV/FFV1/LPCM video files. As I mentioned, I recently started using it for our PDFs and it's a real time saver. I still use DPF Manager for our TIFF files, but it would be nice to use MediaConch across the board. I'm a sucker for uniformity and may explore porting my configuration specs into MediaConch.



File	Implent	Policy	MediaInfo	MediaTrace	Status
out_1925_03.pdf	✓ Valid	✓ PDF is PDF/A	⊕ ⊕	⊕ ⊕	✓ Analyzed
out_1928_05.pdf	✓ Valid	✓ PDF is PDF/A	⊕ ⊕	⊕ ⊕	✓ Analyzed
out_1929_11.pdf	✓ Valid	✓ PDF is PDF/A	⊕ ⊕	⊕ ⊕	✓ Analyzed
out_1934_03.pdf	✓ Valid	✓ PDF is PDF/A	⊕ ⊕	⊕ ⊕	✓ Analyzed
out_1936_01.pdf	✓ Valid	✓ PDF is PDF/A	⊕ ⊕	⊕ ⊕	✓ Analyzed
out_1936_03.pdf	✓ Valid	✓ PDF is PDF/A	⊕ ⊕	⊕ ⊕	✓ Analyzed
out_1936_05.pdf	✓ Valid	✓ PDF is PDF/A	⊕ ⊕	⊕ ⊕	✓ Analyzed
out_1936_11.pdf	✓ Valid	✓ PDF is PDF/A	⊕ ⊕	⊕ ⊕	✓ Analyzed
out_1937_11.pdf	✓ Valid	✓ PDF is PDF/A	⊕ ⊕	⊕ ⊕	✓ Analyzed

Why do you think file validation is important?

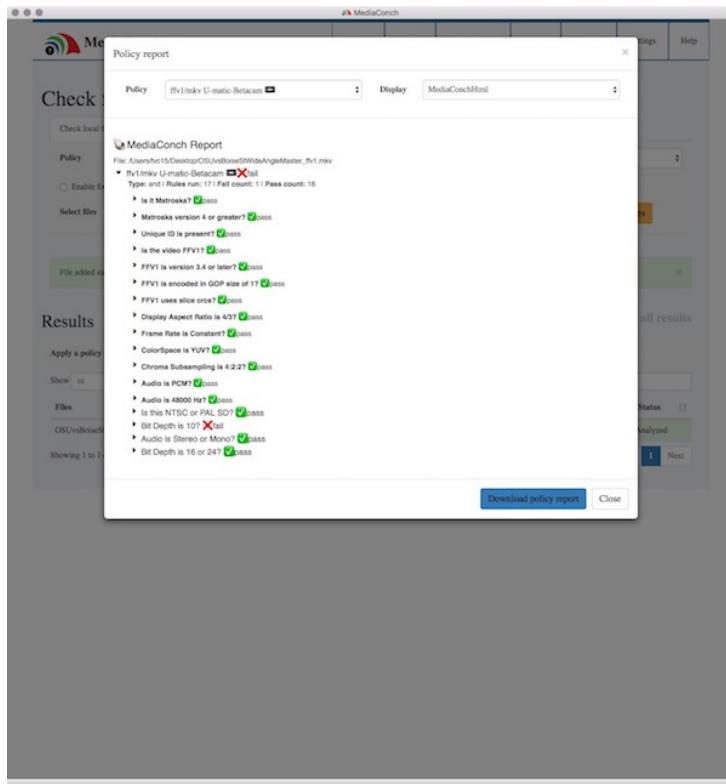
When I first started working in my current position, if a file opened then that was good enough for everyone. But as we all know, there's more to it than that.

I moved to PDF/A-1b for PDFs coming out of our digital production process in 2012. To keep to that particular flavor of PDF, I configured a number of Adobe Acrobat actions for students to ensure that they're saving as PDF/A-1b. There are times when those actions stop working or a student decides to do something else. Whatever the case, the file validation that MediaConch does helps me catch those files early in a project.

I'm not immune to making mistakes either. I literally sit in the middle of a donut of computers and multitask the day away. There are a number of vrecord settings that are easy to mis-select during my videotape transfer process and MediaConch policies are my insurance. Moving from a Betacam SP workflow down to an EP-recorded VHS tape workflow, I could fail to drop the bit depth down to 8. My local policy for low quality VHS ensures that I don't end up pushing a larger 10-bit transfer up to preservation storage. This makes everyone happy, especially the systems people who maintain our storage. Did I mention that they hate me and my big files?

Anything else you'd like to add?

I've been informally testing [Archivematica](#) for close to two years now, primarily on image files that we produce. The machine that I'm running it on is a Mac Pro from 2008 and it chokes a bit on some of the video files, so I haven't done a great deal of testing on those. However, my library is moving forward with a large-scale Archivematica pilot later this year and I'm very much looking forward to trying out the MediaConch integration. That is, if I can convince them to run a development release of it.



The screenshot shows the MediaConch application interface with a 'Policy report' window open. The window displays a list of rules applied to a file named 'f11.mkv'. The file is identified as 'f11.mkv U-matic-Betacam' and is analyzed using the 'all' policy. The results show 11 pass rules and 1 fail rule. The fail rule is 'Bit Depth is 10'.

Rule	Status	Message
File: f11.mkv U-matic-Betacam	Pass	File
Type: and 1 Rules run: 17 / Fail count: 1 / Pass count: 16	Pass	
► Is it Matroska? ✓ pass	Pass	
► Matroska version 4 or greater? ✓ pass	Pass	
► Unique ID is present? ✓ pass	Pass	
► Is the video FFV1? ✓ pass	Pass	
► FFV1 is version 3.4 or later? ✓ pass	Pass	
► FFV1 is encoded in GOP size of 1? ✓ pass	Pass	
► FFV1 uses slice crst? ✓ pass	Pass	
► Display Aspect Ratio is 4:3? ✓ pass	Pass	
► Frame Rate is Constant? ✓ pass	Pass	
► ColorSpace is YUV1? ✓ pass	Pass	
► Chroma Subsampling is 4:2:1? ✓ pass	Pass	
► Audio is PCM? ✓ pass	Pass	
► Audio is 48000 Hz? ✓ pass	Pass	
► Is this NTSC or PAL? ✗ fail	Fail	Bit Depth is 10
► Bit Depth is 10? ✓ pass	Pass	
► Audio is Stereo or Mono? ✓ pass	Pass	
► Bit Depth is 16 or 24? ✓ pass	Pass	