Gregory Helmstetter

Digital Preservation

Final Project

December 15, 2017

# Does That Look Right?

## *Playback Software in Digital Preservation*

With the ubiquity of digital files persisting throughout every aspect of modern life, from cell phone videos to time-based art to video preservation and archiving, one major question is often asked when playing back a digital file: Does that look right? This is an important question for several reasons. First, this is often the first question conservators, preservationists, or archivists ask when playing back video on any platform, whether it be in digital playback software or on analog machines. Second, as of the writing of this paper, I am frankly still questioning whether or not particular digital videos look the way they were intended to look and still questioning how software plays a role in the presentation of videos. And third, the results of this paper and the informal case study I conducted specifically to test playback software may generate more questions that conservators, et al, should ask when playing back digital files. This paper, therefore, will survey a number of playback software tools that archives rely on in quality control processes (particularly as they pertain to large-scale digital preservation workflows); it will consider the critical role that software plays when displaying digital content for preservation and quality control processes and how software is integrated into these workflows; and it will outline and address issues one might encounter when playing digital video files on different software.

## The Case Study

First it will be beneficial to discuss from where this idea originated. The impetus of a survey of digital software occurred during the quality control process of my internship at the NYPL with my supervisor Ben Turkus. As part of a Andrew W. Mellon funded grant, the library has embarked on a large-scale digitization project in which they are sending various formats of magnetic media (both audio and video) to vendors for reformatting. The original file specifications that vendors had to adhere to include sending preservation master files in a v210 uncompressed video codec wrapped in the QuickTime file format with 10-bit bit depth and chroma subsampling of 4:2:2 YUV (In the past few months, they have since changed the specifications to FFV1 version 3 wrapped in matroska, 10-bit, 4:2:2 YUV).[1] As part of their quality control process, NYPL checks metadata fixity using JSON validation and then conducts manual quality control on their video files. When I was conducting quality control on v210 uncompressed QuickTime files, Turkus suggested using multiple playback software programs to look at the files and to see how, if at all, they displayed video differently. In our side-by-side comparison, two we noticed that the digitized videos were in fact displayed differently, both in terms of aspect ratio and saturation. An example of this can be seen in Image 1 below.

---

[1] *AMI Digital Asset Technical Specifications* (2017; repr., The New York Public Library, 2017), https://github.com/NYPL/ami-specifications.


Image 1. Four different playback software tools displaying the same video file differently.

As can be seen in the above image, the same v210 uncompressed file is displayed in different ways, depending on the software being used to playback the file. The four programs we used for playback was QuickTime 7.6.6, QuickTime 10.4, MPV 0.27.0, and VLC 2.2.6, and each one was interpreting the file in slightly different ways. First, the files was opened in a different size in each, but they all maintained the correct aspect ratio, except one QuickTime 7. As can been seen in on the bottom left of Image 1 above, the person's face is slightly squashed compared to the other displays. Another difference is the saturation of skin color. MPV and VLC (on the right in Image 1) display a more saturated version of the video file than QuickTime 7 and 10 on the left. These were surprising differences, and with this survey, the intent is therefore aimed to understand why each software program displays the files differently and how to correct and/or compensate for these differences.

## The Software

For the most part, we used QuickTime 7.6.6, QuickTime 10.4, MPV 0.27.0, and VLC 2.2.6 in this survey because these are the tools used at NYPL for their quality control process, they are popular with both the archival community and consumers, and they represent a balance between proprietary and open source software, but the history of each program reveals other important reasons for their ubiquity. The first version of Apple's QuickTime was released in 1991, developing over time through the addition of different features and the ability to playback various file formats.[2] In 2005, the first version of QuickTime 7 was compatible with Mac Snow Leopard and OS X Lion (there are versions that are available for Windows as well).[3] Quicktime 7 used QTkit as its underlying media framework, which is essentially what allows software to playback media, until it abandoned QTkit in 2009 in favor of the AVFoundation framework for the newer version, QuickTime 10.[4] The VLC media player is free and open-source software that came out of the the VideoLAN project, which began in 1996 by a group of

[2] "QuickTime: History," *Wikipedia*, November 11, 2017, https://en.wikipedia.org/w/index.php?title=QuickTime&oldid=809815150.
[3] "Download QuickTime Player 7 for Mac OS X V10.6.3 or Later," accessed December 15, 2017, https://support.apple.com/kb/DL923?locale=en_US.
[4] "Technical Note TN2300: Transitioning QTKit Code to AV Foundation," accessed December 15, 2017, https://developer.apple.com/library/content/technotes/tn2300/_index.html.

students, and has been a non-profit since 2009.[5] VLC is modular based like QuickTime and uses Cocoa API which includes the frameworks Foundation Kit, Application Kit, and Core Data.[6] MPV is another free, open-source software that originated on GitHub from mplayer and uses its framework.[7] As can be seen, each playback software has originated from different auspices to achieve different objectives, and it should come as no surprise that they were developed using different frameworks which are essential to how each program displays video files.

## Playback Problems and How to Solve Them

In his article for the Tate, Dave Rice, archivist at City University of New York, identifies three main approaches for ensuring consistent video playback: emulation, normalisation and migration. In emulation he talks about how to maintain or recreate playback software to ensure that a video intended to play on a specific application can continue to be be playable. In normalisation he explains the need for reformatting files to accommodate for changes in software, and in migration, it is required to document changes in presentation to ensure proper playback in the future.[8]

In our NYPL example, an emulation problem exists that prevents us from playing the v210 uncompressed file in QuickTime. The main cause of this problems is the framework, or the architecture that makes up the base of the software, and this is where some of the differences in the playback of the file originate. The framework allows for

---

[5] "VideoLAN, a Project and a Non-Profit Organization," accessed December 15, 2017, https://www.videolan.org/videolan/.
[6] "Cocoa (API)," Wikipedia, October 22, 2017, https://en.wikipedia.org/w/index.php?title=Cocoa_(API)&oldid=806510376.
[7] "MPV, A New Fork Of MPlayer/MPlayer2 - Phoronix," accessed December 15, 2017, https://www.phoronix.com/scan.php?page=news_item&px=MTQyODQ.
[8] Dave Rice, "Sustaining Consistent Video Presentation," Tate, accessed November 26, 2017, http://www.tate.org.uk/about-us/projects/pericles/sustaining-consistent-video-presentation.

different video formats, audio formats, and transmission protocols (among other elements) to integrated into the software and allows different videos to play.[9] The framework calls upon different components or libraries that include specifications for which formats can be played back, and additional components and libraries are available as plugins to offer the ability to play other codecs not native to the framework. QuickTime uses components to playback files, while VLC and MPV use libraries, and the mere fact that different software calls upon different sources (i.e., libraries or components) for the playback of files is a contributing factor for why we see differences in file presentations.

As far as components are concerned, this is one of the major hurdles that we ran across in our test and my attempt at duplicating the differences. As Dave Rice explains, "Sustaining video presentations through emulation requires maintaining a player and all of its dependencies. For instance, if a creator determines that a video is intended for presentation through QuickTime Pro 7 this may mean preserving QuickTime Pro 7 as an application along with its underlying set of components as well as an operating system that can support QuickTime Pro 7's underlying 32 bit QtKit framework."[10] In our effort to playback the v210 uncompressed file on QuickTime 7, only the audio would play, which was attributed to the fact that the v210 format is not natively supported by QuickTime 7. Also, since Apple no longer supports QuickTime 7, the proper component could not be downloaded from Apple's website. This is one issue that archives and conservators will face in the future when trying to use this software. As newer operating systems appear on the market, they will eventually stop supporting QuickTime 7 and

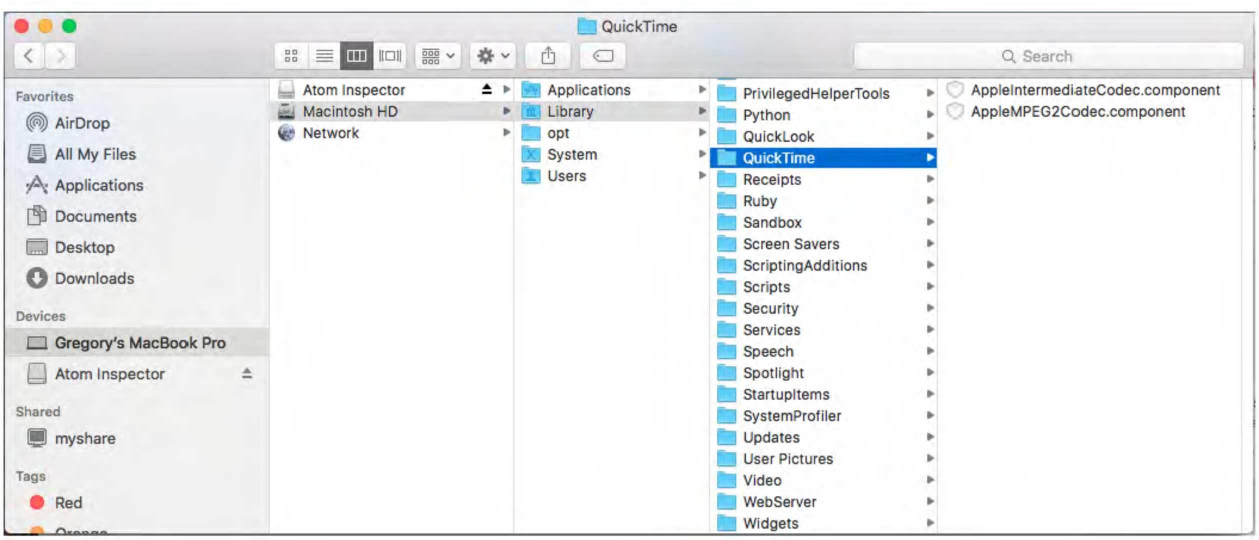
---

[9] "Multimedia Framework," Wikipedia, July 20, 2017, https://en.wikipedia.org/w/index.php?title=Multimedia_framework&oldid=791394254.

[10] Rice, "Sustaining Consistent Video Presentation."

emulation (or finding applications to imitate the functions of QuickTime) will be required.

As Rice also points out, other software applications download other software into the folders in which QuickTime components are stored.[11] This is one form of emulation and can affect the way files are played back in QuickTime, either for the better or worse depending on the situation. In the NYPL test case, this was a way to overcome the challenge we were facing, the inability to play v210 files. On my computer, QuickTime components are stored in the /Library/QuickTime/ folder, and it is evident in Image 2 that there were not the correct components to play the files back. In order to remedy the issues, Turkus recommended installing the Blackmagic Desktop Video 10.9.7 which adds components to the QuickTime Library (see Image 3.) that enable the software to playback v210 uncompressed files. As Rice further notes, "the [library] files used by VLC are from within the VLC application itself with a much lower reliance on system files," which means that to maintain and emulate VLC as it has older versions of the software on its website.[12]




[11] Ibid.
[12] Rice, "Sustaining Consistent Video Presentation."

Image 2. QuickTime Library before installing Blackmagic Desktop Video 10.9.7

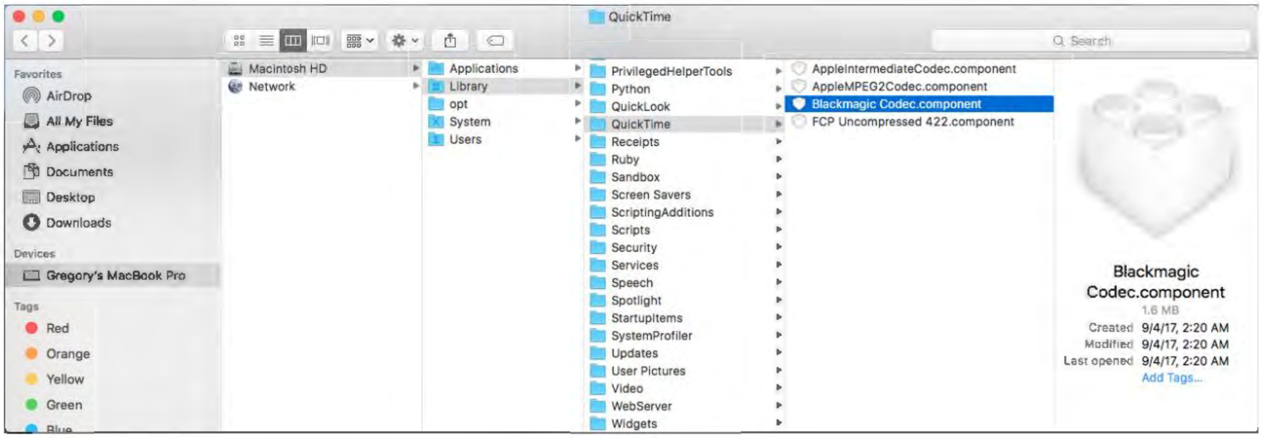



Image 3. QuickTime Library after installing Blackmagic Desktop Video 10.9.7 which includes components that enable playback of v210

In the same way that software requires components or libraries to playback a file, a file must contain appropriate atoms that complement the software. Atoms are the basic building blocks of any file format and are structured in a similar way as any other file format,[13] like video and audio codecs contained within a wrapper.[14] Specific atoms dictate specific characteristics of a file like aspect ratio and color space, which are two addition issues occuring in our NYPL example. As is evident when playing the v210 uncompressed file on QuickTime 7 and VLC, there was a noticeable difference in the hue or saturation of the skin tones (Image 4, below). This same difference was also observed when comparing QuickTime 10 to VLC, as well as both QuickTime versions to MPV,

[13] "Overview of QTFF," accessed December 10, 2017, https://developer.apple.com/library/content/documentation/QuickTime/QTFF/QTFFChap1/qtff1.html#//apple_ref/doc/uid/TP40000939-CH203-BBCGDDDF.

[14] Joshua Ranger, "A Primer on Codecs for Moving Image and Sound Archives," AVPreserve (blog), February 19, 2013, https://www.avpreserve.com/a-primer-on-codecs-for-moving-image-and-sound-archives-2/.

which makes sense. Both versions of QuickTime display the color the same, while VLC and MPV display color the same way.

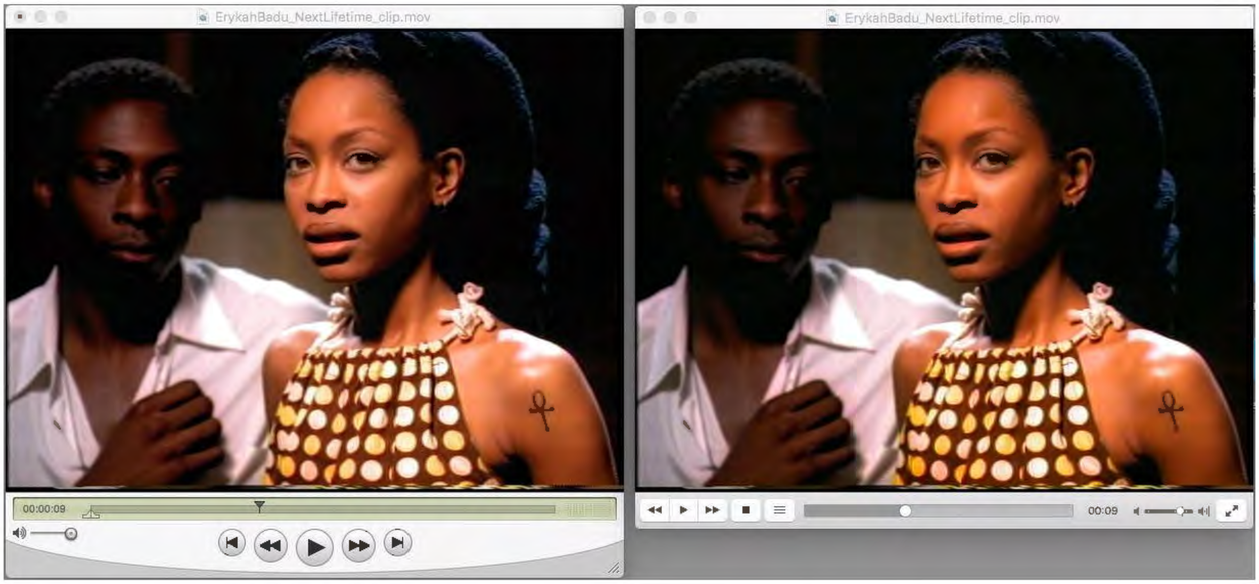



Image 4. The difference in how color is displayed in QuickTime 7 and

Like so much of the differences that occur in playback, it is difficult to understand exactly what is going on within the playback software, but in short, the problem relates to the way in which the color from the file format is being color matrixed or converted when it played back by the software. The v210 uncompressed file originated from a tape standard definition magnetic format (e.g., Betacam) and its metadata indicates that it is encoded as YUV 4:2:2 (or rather a YCbCr in the broadcast community), meaning that is was captured using the BT.601 standard defining standard definition video.[15] When the file is played back digitally, the software changes the way that information is interpreted

[15] "Rec. 601," *Wikipedia*, November 19, 2017, https://en.wikipedia.org/w/index.php?title=Rec._601&oldid=811076905.

because computers use an RGB colorspace or rather it will display the video using the BT.709 standard.[16]

As Rice notes, the QuickTime container stores this color information in specific atoms, and if the atom is not present, QuickTime will default to the BT.601 standard.[17] However, if the color information is present in the file, it will play to the standard present in the file. When we inspecting the metadata presented by MediaInfo (Image 5, below) for the v210 uncompressed file from NYPL, it is clear that the file is specified as BT.601. This means that QuickTime 7 and 10 are playing the file back on the BT.601 standard, which is the file's intended standard, and we can infer that this difference between the way VLC and MPV display the file is due to VLC and MPV matrixing (i.e., converting) the video to the BT.709 standard and displaying the file in the high definition colorspace. Therefore, the intended color presentation is closest to the way the QuickTime 7 and 10 play back the file.

[16] "Rec. 709," *Wikipedia*, December 7, 2017, https://en.wikipedia.org/w/index.php?title=Rec._709&oldid=814229789.
[17] Rice, "Sustaining Consistent Playback Presentation."

```
Video
ID                                       : 1
Format                                   : YUV
Codec ID                                 : v210
Codec ID/Hint                            : AJA Video Systems Xena
Duration                                 : 25 s 394 ms
Source duration                          : 25 s 425 ms
Bit rate mode                            : Constant
Bit rate                                 : 224 Mb/s
Width                                    : 720 pixels
Clean aperture width                     : 720 pixels
Height                                   : 486 pixels
Clean aperture height                    : 486 pixels
Display aspect ratio                     : 4:3
Clean aperture display aspect ratio      : 4:3
Frame rate mode                          : Constant
Frame rate                               : 29.970 (30000/1001) FPS
Standard                                 : NTSC
Color space                              : YUV
Chroma subsampling                       : 4:2:2
Bit depth                                : 10 bits
Scan type                                : Interlaced
Scan type, store method                  : Interleaved fields
Scan order                               : Bottom Field First
Compression mode                         : Lossless
Bits/(Pixel*Frame)                       : 21.333
Stream size                              : 677 MiB (99%)
Source stream size                       : 678 MiB (99%)
Language                                 : English
Encoded date                             : UTC 2016-10-27 20:30:20
Tagged date                              : UTC 2016-10-27 21:16:14
Color primaries                          : BT.601 NTSC
Transfer characteristics                 : BT.709
Matrix coefficients                      : BT.601
```

Image 5. MediaInfo for the NYPL v210 uncompressed file.

Morgan Morel, the lead archivist at the Bay Area Video Coalition (BAVC), sums up the dangers that this difference between QuickTime and VLC presents for video preservation, saying “The big fear is that a playback system would ‘fix’ an error with the file without telling the user. This is why we generally avoid using players like VLC, which will playback just about anything. Quicktime 7, however finicky, is at least consistent with its problems. Also, we don’t want a player that makes decisions about the playback without telling the user. These decisions could be the playback aspect ratio, color space,

[and] frame size."[18] The leads to another major issue that we experienced regarding the aspect ratio of the v210 uncompressed file from NYPL. Specifically, when opening the file in each payback software, only QuickTime 7 manipulated the aspect ratio.

When looking at the file metadata, extended specifications can be seen using the MediaInfo -f tag, which displays two important pieces of metadata: the "pixel aspect ratio" and the "clean aperture ratio" (see Image 6, below). The pixel aspect ratio is 0.900 which means that the pixel is not a perfect square but rectangular, and when opening the file in QuickTime 7, the file looks flattened. In QuickTime 7, the way to adjust fix this issue is by using the keyboard shortcut "Command + J" which displays the file properties and navigating to the "Presentation" tab (Image 7). There you will see a setting called the "Conform aperture to" dropdown. This is automatically set to the "Classic" setting which displays the video using a 1:1 or square pixel aspect ratio, but because the file has the 0.900 aspect ratio, the image is manipulated. To remedy this, select the "Clean" aperture setting which tells QuickTime to set the video file to the metadata element called the "Clean aperture pixel aspect ratio" (see Image 6). This "quirk" may be one of the reasons that archivists like Morel and Turkus prefer using QuickTime 7 over QuickTime 10 because it gives the some amount of control as to how a file's aspect ratio is presented. Overall these

[18] Morgan Morel to Gregory Helmstetter, "BAVC/MIAP Quality Control Intro," December 8, 2017.

```
Width                                    : 720
Width                                    : 720 pixels
Clean aperture width                     : 720
Clean aperture width                     : 720 pixels
Height                                   : 486
Height                                   : 486 pixels
Clean aperture height                    : 486
Clean aperture height                    : 486 pixels
Pixel aspect ratio                       : 0.900
Clean aperture pixel aspect ratio        : 0.900
Display aspect ratio                     : 1.333
Display aspect ratio                     : 4:3
Clean aperture display aspect ratio      : 1.333
Clean aperture display aspect ratio      : 4:3
Rotation                                 : 0.000
Frame rate mode                          : CFR
Frame rate mode                          : Constant
Frame rate                               : 29.970
Frame rate                               : 29.970 (30000/1001) FPS
FrameRate_Num                            : 30000
FrameRate_Den                            : 1001
Frame count                              : 761
Source frame count                       : 762
Standard                                 : NTSC
Resolution                               : 10
Resolution                               : 10 bits
Color space                              : YUV
Chroma subsampling                       : 4:2:2
Chroma subsampling                       : 4:2:2
```

Image 6. MediaInfo -f for the NYPL v210 uncompressed file.

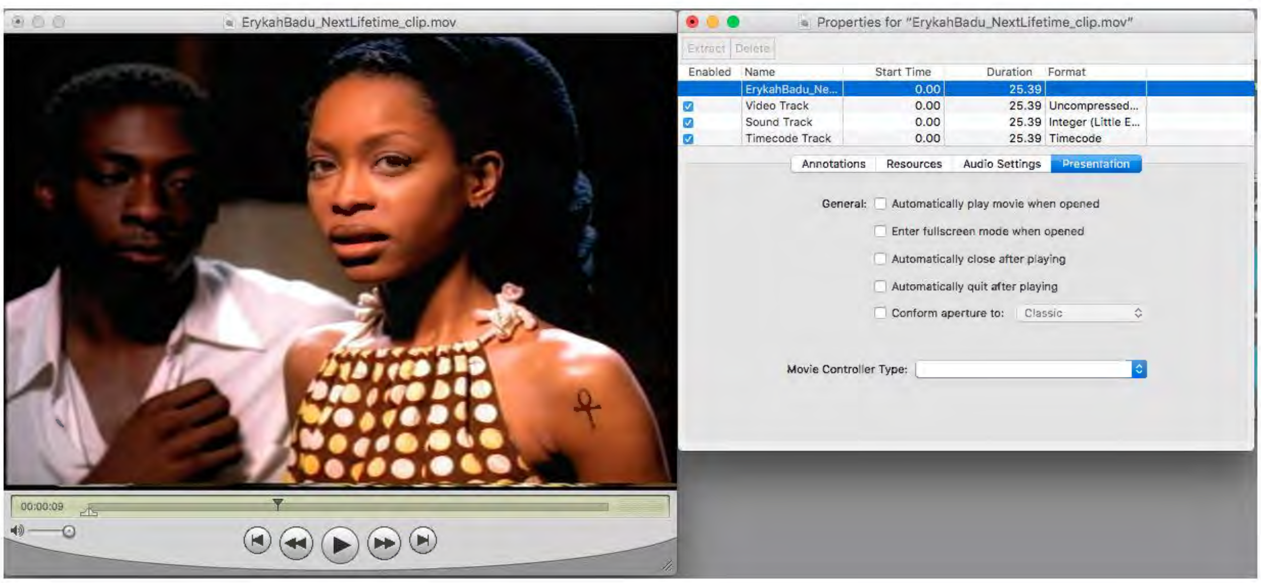



Image 7. The properties tab appears when using the keyboard shortcut Command + J

## Quality Control Considerations and Conclusions

The problems outlined above are only a few examples of the integral role that playback software plays in a digital preservation workflow, and it is important to understand the differences that may occur between different software. In addition, each institution may us the software for a variety of different reasons and may have different goals in mind for software. Some potential uses of the software outlined above include emulation, normalisation, and migration as three different uses of software and why it is important to either maintain the software or maintain the properties of the file so that content can be played back as true to its original form as possible.

Quality control is another important role of playback software which has only been touched upon up to this point. There are a number of tools that can be used in tandem with playback software to ensure that files are being presented properly. MediaConch is one piece of software that checks the metadata of a file to ensure that it conforms to an institutions faile specifications, and QC Tools is an open source quality control program that allows for "the inspection of video signal characteristics for batches of digital media, in order to prioritize archival quality control, detect common errors in digitization, facilitate targeted response, and thus increase trust in video digitization efforts."[19] In addition, Morgan Morel highlights the importance of a good balance between using automated tools like MediaConch and QC Tools with manual processes like good, old-fashion human observation (i.e., watching the files) to provide the best result in ensuring that a file plays back in its intended form. Morel says,

---

[19] Kelly Haydon, "QCTools," Text, BAVC, October 8, 2015, https://www.bavc.org/preserve-media/preservation-tools/qctools.

“Automated QC should be used to find issues that humans are not good at, but that computers are. Comparing hundreds of metadata fields is something that computers are very good at. Comparing the overall image quality of a video transfer is something that computers are not good at. A healthy mix of automated and manual is necessary for a good QC workflow.”[20] This approach requires archivists and conservators to consider the pros and cons of a particular playback software while not becoming beholden to it.

As can be seen in the case study examined in this paper, various issues can present themselves and it is up to human beings to understand the equipment they are using. Some of the problems occuring with color space and the aspect ratio, in particular, are sometimes difficult to discover, and it is important to have other tools at your disposal. It is important to build in effective quality control processes because as Morel puts it one last time, quality control is “constantly a moving target, because we need to continue trying to find issues, then try and find ways to stop those issues from happening. In a sense, quality control should be about root-cause analysis. We need to identify issues, identify the root of the issues, and then stop the root of the problem from happening.”[21]

## Bibliography

“A Short Guide to Choosing a Digital Format for Video Archiving Masters | Scart - a Website on Audiovisual Heritage by Packed.” Accessed December 4, 2017. https://www.scart.be/?q=en/content/short-guide-choosing-digital-format-video-archiving-masters.

[20] Morel to Helmstetter, “BAVC/MIAP Quality Control Intro.”.
[21] Ibid.

*AMI Digital Asset Technical Specifications*. 2017. Reprint, The New York Public Library, 2017. https://github.com/NYPL/ami-specifications.

"Apple - QuickTime - QuickTime Player - Tech Specs," July 24, 2008. https://web.archive.org/web/20080724002332/http://www.apple.com/quicktime/player/specs.html.

Chan, Glenn. "'HD' versus 'SD' Color Space." Accessed December 10, 2017. http://www.glennchan.info/articles/technical/hd-versus-sd-color-space/hd-versus-sd-color-space.htm.

"Cocoa (API)." *Wikipedia*, October 22, 2017. https://en.wikipedia.org/w/index.php?title=Cocoa_(API)&oldid=806510376.

"Comparison of Video Player Software." *Wikipedia*, December 1, 2017. https://en.wikipedia.org/w/index.php?title=Comparison_of_video_player_software&oldid=813001440.

"Download QuickTime Player 7 for Mac OS X V10.6.3 or Later." Accessed December 15, 2017. https://support.apple.com/kb/DL923?locale=en_US.

Haydon, Kelly. Interview about Playback Software and Quality Control. Interview by Gregory Helmstetter, December 7, 2017.

Haydon, Kelly. "QCTools." Text. BAVC, October 8, 2015. https://www.bavc.org/preserve-media/preservation-tools/qctools.

"LibVLC Media Player, Open Source Video Framework for Every OS! - VideoLAN." Accessed December 12, 2017. https://www.videolan.org/vlc/libvlc.html.

Morel, Morgan. Email to Gregory Helmstetter. "BAVC/MIAP Quality Control Intro," December 8, 2017.

"MPlayer - The Movie Player." Accessed December 12, 2017. http://www.mplayerhq.hu/design7/info.html.

"MPV, A New Fork Of MPlayer/MPlayer2 - Phoronix." Accessed December 15, 2017. https://www.phoronix.com/scan.php?page=news_item&px=MTQyODQ.

"Multimedia Framework." *Wikipedia*, July 20, 2017. https://en.wikipedia.org/w/index.php?title=Multimedia_framework&oldid=791394254.

"Overview of QTFF." Accessed December 10, 2017. https://developer.apple.com/library/content/documentation/QuickTime/QTFF/QTFFChap1/qtff1.html#//apple_ref/doc/uid/TP40000939-CH203-BBCGDDDF.

"QuickTime." *Wikipedia*, November 11, 2017. https://en.wikipedia.org/w/index.php?title=QuickTime&oldid=809815150.

"QuickTime - QuickTime Components - Apple (香港)." Accessed December 10, 2017. https://www.apple.com/hk/quicktime/resources/components.html.

Ranger, Joshua. "A Primer on Codecs for Moving Image and Sound Archives." *AVPreserve* (blog), February 19, 2013. https://www.avpreserve.com/a-primer-on-codecs-for-moving-image-and-sound-archives-2/.

"Rec. 601." *Wikipedia*, November 19, 2017. https://en.wikipedia.org/w/index.php?title=Rec._601&oldid=811076905.

"Rec. 709." *Wikipedia*, December 7, 2017. https://en.wikipedia.org/w/index.php?title=Rec._709&oldid=814229789.

Rice, Dave. "Sustaining Consistent Video Presentation." Tate. Accessed November 26, 2017. http://www.tate.org.uk/about-us/projects/pericles/sustaining-consistent-video-presentation.

Singer, Martha. *TechFocus III: Caring for Software-Based Art - Agathe Jarczyk*, 2015. https://vimeo.com/149089331.

“Technical Note TN2227: Video Color Management in AV Foundation and QTKit.” Accessed December 15, 2017. https://developer.apple.com/library/content/technotes/tn2227/_index.html.

“Technical Note TN2300: Transitioning QTKit Code to AV Foundation.” Accessed December 15, 2017. https://developer.apple.com/library/content/technotes/tn2300/_index.html.

“The Future of Video Player Accessibility: Q&A Highlights.” 3Play Media, April 28, 2015. http://www.3playmedia.com/2015/04/28/the-future-of-video-player-accessibility-webinar-qa-highlights/.

“VideoLAN, a Project and a Non-Profit Organization.” Accessed December 15, 2017. https://www.videolan.org/videolan/.

“VLC Media Player.” *Wikipedia*, December 12, 2017. https://en.wikipedia.org/w/index.php?title=VLC_media_player&oldid=815108933.

“Webinar: To Standardise or Not: Is Normalisation of Video a Thing of the Past?” PrestoCentre, August 13, 2014. https://www.prestocentre.org/webinar-standardise-or-not-normalisation-video-thing-past.