

#NoTimeToWait 19 July 2016

Matroska working group meeting notes

End-of-day shareout (tl;dr):

- Can Matroska meet the needs of archivists in its current form?
- Some aspects can meet some of the needs (don't need to extend) but other things -- time codes, etc
- Fitting time codes into the subtitle stream (or not)
- How can we take existing metadata info and fit it in -- METS/MODS, etc. into a tag form or other?
- Ambisonic and 360* virtual reality video mappings, mapping onto a plane or sphere -- not specified

Group leader: Michael Bradshaw

Group absorber: Steve Lhome

Group note-taker: Ashley Blewer

Going round the room getting interest and engagement:

Hermann Lewtitz - Interested in DVD

Nathaniel: Preservation

Steve Lhome - Created Matroska!!!

Michael Bradshaw -- aligning webm with matroska, for people creating parsers

Tobias Rapp (NOA GmbH) -- ffv1+avi, now goals are to find out specifics of the container

Albrecht Wiedmann - from museum around the corner, small sound and video archive and starting to digitize the materials

Ashley Blewer -- MediaConch

Kate Murray -- Library of Congress, has list of questions to see what support for different aspects are and how they compare to AS-07, how Matroska does things differently

Marion Jaks - Thinking about changing from avi to Matroska, thinking about DVD preservation

Talking about Tobias's talk from yesterday, Steve is saying he was unfamiliar with an "ingest report" and need for that.

Tobias: When looking at Matroska, it might make sense to look at a standard like PREMIS and have it as an attachment, either by name or another identifier, to have the ingest report.

Kate brings up REVTMD as something created at the National (US) Archives. The schema, an old version, is online at <http://archives.gov/preservationproducts/revtmd> Took audio and transcribed it for video. Something Matroska might be interested in. The schema itself is moving and hosted at Harvard. Says people like the schema because of the process history chunk. The chunk covers audio and video.

Steve asks if the audio/video can be separated, so can you have both distinctly? Kate says yes, they can be separated out if that's preferred.

Tobias mentions it being like film archiving, film is scanned separately from audio and joined later.

Hermann brings up quality markers, and they exceeded the limit. Later on, there was a limit for the header which in some cases -- not all information would fit inside of the header. When looking at different cases -- example, capturing audio and video information in a different way -- it's much more info to store. Better to have a solution with no limits (size limits).

Steve emphasizes that there's no size limit in Matroska. There's a VOID element so even if info is at the beginning of the file and there was no more room -- it can follow along in the back of the file.

Ashley: The limit is 72TB!

Steve says he'd like to work on that for the specs since it's something that is needed. One format can be as it is, and one format solution where the information is merged into something more EBML-like. The schema, any schema can probably map very easily and have something more Matroska-like so that any format can merge into it. "We'll see when we discuss the technical details."

Hermann asks Kate, Is Specification AS-07, would it be possible to take the spec and adopt it to Matroska?

Kate says it's more descriptive in nature, about identifiers and titles. Any parametric information goes into sound/picture SMPTE3771 identifiers. Depends on the metadata in question. No space for coding history.

Hermann says his understanding of AS-07 is to define the requirements of metadata and doesn't think it makes sense to make a spec for MXF and ffv1. If choosing ffv1, would also choose Matroska. But says people want it all in one file and have spent time and money for the spec, to design the spec. It'd be a good thing to take this work and to define the spec for Matroska. Using AS-07 information. Is it possible to adapt it? All the defining work already done... Most people don't care if the ending is MXF or MKV, they just want the features.

Kate says sure, there's no reason why you couldn't map it to MKV. Depends on the specific parts. AS-07 composed of DMSs quite easy to map, any other metadata in any format you want would be held in the Generic Stream partitions, the MAVIS/METS record, etc, could go into that stream partition.

Ashley asks about the general stream in Matroska, metadata going into there.

Steve says you can tag and point to each track, time data is within chapters. Defining parts of the file, you can tag just that part you're defining. So you can do a lot of stuff with tags, it's very flexible. Steve talks about someone doing tag mapping and the idea for Matroska was that whatever existed elsewhere can be mapped into the system. And there is no holes. And hopes it's still the same for other system tags.

Just a matter of mapping something new into the system.

Michael asks about the Partial File Retrieval noted in Tobias's slides.

Tobias says if you have a very large file, very long, and you only want a segment of it... only parts might be needed, customers spend money on the servers that have features that allow them to grab partial file. Binary of "this plus this" extracted and reconstructed. Fast delivery from the tape, you don't need to read ten tapes

Ashley: You don't need the beginning and the end, you can just pick out the segment that you want.

Hermann confirms.

Michael: Don't you still need the header?

Tobias: That's the trick. One solution for avi is there's a separate db with indexes, seeking table as a level on top of the storage system. When a request is made, you can be like "this time to this time" and they can say the best seeking position is this+that, so the database is used.

Michael: YouTube does this when needed to watch a video, there's info about the codec and a seeking table at the beginning of the file... but could also be stored at a separate db. Fetch the beginning segment. Maybe someone wants 3m into the video, that tells us where the 3m mark is (the table) so we get chunks from that forward. The data comes from that point.

Tobias: Same as web seeking, same concept.

Michael: You can def fetch that, fetch or store the header and know where to seek to, but the info where to seek -- it's definitely there to grab sub-portions of the file.

Tobias and Steve discuss time. Steve says when there are reference frames in the future, they are put in the bitstream before so all the info is there in time. But it's a precise order, not random wherever you want it.

Tobias references FFV1 use case.

Michael: Monotonically-increasing in that case.

Steve says one of the core designs were made for this, streaming. At first you may download just a very small piece to read anything and go anywhere in the file and even if you seek anywhere and it's not in the right position, you can always get back in your feet.

Mapping from AVI spec, this was discussed with Dave Rice recently and how to bring it back into MK. Steve says he wasn't confident in using the AVI system since from the beginning, wasn't sure if there wasn't something better/ more general. Like maybe it's not "good enough" (references Ambisonic (sp)). "We want that feature but don't know the right format to do it, want something to be future-proof."

Tobias: AVI, we have situation of digitizing tapes with 8 channels, don't know what's on the channels. Mono, etc?

Steve says there's not really a way in MXF to define a position.

Tobias: In AVI we set an empty channel, there's no way to set it in the stream. But general section in AVI is maybe not sufficient.

Steve: channel mapping is defined for one channel, what's inside that channel, and it doesn't cover mapping for each channel to play together. In MKV there is way to link different channels, designed for 3 videos -- one encoded one track, one in another, something in MKV that says we have to play this together. And this is left, this is right. Same system could be for audio as well. Not in there yet.

Michael: There's ambisonic channel mappings in the IETF Opus stream, might be a good place to look. Different types of formats, too. Opus, just using B-WAV

RFC available here: <https://tools.ietf.org/html/draft-graczyk-codec-ambisonics>

Ashley How long has Opus been in the process of standardization?

Opus itself is standardized but extending it because just uses channel mapping from vorbis for normal mono-stereo-7.1 surround. Opus incorporated that and left 254 channel options undefined. So just adding a new proposal for channel mapping for round 2.

Kate says AS-07 does have a complicated system about labelling audio channels and other layout details. This is something that could be mapped over.

Michael: Channel positions... when you just want to play video, you want it to be v. simple but sometimes where if doing professional theater you don't want it just be simple and you want it to be precise

[brief interruption in notes stream]

Topic is on YUV, RGB. Audio requirements?

Michael says Apple channel layouts you can specify position of each speaker, for 3D space. If needed that could work. But if you just want a simple file... it would be overkill. Might be hard to figure out what position in 3D space does this speaker correspond to?

Steve: With EBML, using default value you don't have to write anything. If requirement is 24 channels you can use the system.

Michael: If we use channel mapping I'd like to have a nice, normal default. Opus, it's not always left-first right-second.

Steve says it's defined at the codec-level so it's not a problem in that case, nor multichannel. Probably FLAC has it?

Tobias says there's nothing in the codec itself for mapping but seen in ffmpeg where just storing the avi channel alongside the file, reconstruct it.

Nathaniel: Understanding as a user? Where is the metadata stored and more importantly how is it stored, what software, where? I'm a beginner of this, it's important for us to use METS/MODS. What's your experience with that, can we get it into the file? Automatically?

Steve says if you can export into XML, you can write it right now using MKVmerge (<https://mkvtoolnix.download/doc/mkvmerge.html>) that's the main program to do it (written by Moritz Bunkus other designer of MKV). It will write it into the file. Don't know how it does it specifically but guidelines for metadata, try to put it at the beginning and if you need to edit it later, either there is space reserved (usually the case for mkv merge). If not room, it goes into a new file

(not great for preservation)

Steve says but it can just go at the end as well!

There is a command line and GUI. And can edit chapter/tags, take source file and there's a main muxing program. Not just for MKV but other formats work as well.

Can go into the tagging system of the file...

Dave Rice is going to turn this into a standardized XML file for generating accurate

Kate says where does that go?

Steve says XML turns into EBML and stored in the MKV form. Not the XML file kept as is but transformed. There's also attachments so a very proprietary format can just go in as an attachment. Tag as attachment and describe it.

Steve says the primary use for attachments is fonts, so people can read with the correct font.

Ashley tries to clarify how fonts work with diverse language sets

Steve says there are security issues with attachments, especially fonts can't be installed yourself because of security risk. So the system might not allow you.

Ashley trying to understand attachments better

Steve: You can attach whatever you want, you can attach something. It knows about the MIME type and it's not up in the air. Steve speaks on systems potentially rejecting attachments if they are deemed unsafe.

Ashley is asking about interoperability and if a potential system-rejecting attachment in the future would break the file.

Steve says the use case right now is mostly fonts and cover art.

Steve: Very little needed for -- chapters, tags attachments they are just extras and not needed for playback. Even if removed from the file or destroyed, the file will still play. And as long as the track info is there at the beginning, you can destroy the other parts

We move on towards Tobias's next slide talking about active image area examples.

Steve: If block based and lossless, you can prob remove that part losslessly, depends on ffv1?

Steve: Matroska has 3 parts to describe the pixel area

<https://matroska.org/technical/specs/index.html#PixelWidth>

In the pixel, there's a problem with other codecs that don't like it. So you define pixel size (bigger than what can be seen), PixelCrop, and DisplayArea mostly for aspect ratio. Different order than (Tobias's slide). Crop defines the cropped area. Display Size is where you describe the pixel aspect ratio -- square, 16:9, 4:3. All the info is taken in a different order.

Marion: From preservation standpoint, it's important to know the original source, not the active image area as much (If letterboxed as it comes in, it's letterboxed)

Michael: Is it important to include in the display area -- change the crop value?

Marion: Depends on what you want, for us we would include the black area because the original is.

Hermann: the pure black wouldn't cost very much

Steve: If you have an analog source, the black isn't black, it is full of noise!

Hermann: I do not speak about LOSSY!

Ashley: Names off the related elements

<https://matroska.org/technical/specs/index.html#PixelWidth>

Steve: DisplayUnit is in cm or inches, for art installations so how the artist intended it to be -- big screen or small screen. Something interesting.

Topic moving to color spaces...

Michael: Won't handle the log functions or testing functions that aren't even described parametrically but rather updated. That wants to be in the ffv1 level and not the mkv. Suggests a push for ICC profile? Codec level and Matroska level -- don't want to dupe too much but what MKV has is currently sufficient. But important to note it doesn't handle every use case and for people who have different colorspace issues.

Something more specific might have to be at the codec level.

Steve says if it involves film, something like the film chemicals, etc.

Steve moves on to DVD preservation. Basically taking the video-audio out and chapters, it's very easy to do. The dvd menu and commands that come with a dvd, that is something he has done as PoC trying to extract all the info from .ifo and .vob and turns it into something MKV-compliant. Even there are games to play within DVDs, virtual machines. It works in VLC but it's not really complete, it would need more work to be a standard, like a spec for matroska.

Michael: Benefit of using MKV instead of an ISO?

Steve: goal was to be able to encode it in something lighter, for preservation there is not really a use case. The only thing is that the dvd spec, you have to pay for them. Players have to pay a license to be able to, you cannot use the system for free. Didn't get the actual spec, it's from info from the internet trying to understand without the spec. From that, managed to make it work. Same as in VLC now.

Ashley: All reverse engineering?

Steve: Yeah trying to find what works and make it work. For preservation, maybe ISO is enough.

Hermann: We have an auto workflow to make web copies but the workflow demands a file. ISO is not...

Marion: How to make it as one file, we are thinking about concatting the mpeg file but if you could handle this through a container, all the complexity of the menu, this would be very interesting.

Steve: that was my goal! Reencode a dvd, make it small. Learn about DVDs and keep the experience. If you want to transmit it over the web...

Hermann: There's a menu... several pages...

Steve: Probably we will remove some parts during specification, the menus are a good candidate because they are a rarely used feature. We will rm some parts of the spec and another specification. Even each codec right now... I think Dave put them in the spec... on our website.

Steve: The tool I made, it separates every piece out. The video, audio, chapters. MKVmerge assembles them in a script I wrote. Before you run, you can re-encode video into a different format. Only then you can run mkvmerge. It's in the specs.

Steve says it's in the specs (Here are the specs:
<http://matroska-org.github.io/matroska-specification/chapters.html>)

Hermann: If I check it and its working, the file I got will follow the specs?

Ashley: It's still compliant.

Hermann: The risk is in processing.

Michael: The MKV file adheres to MKV specification.

Hermann: So it is a limited risk. We make a check afterwards and then we have a serious preservation element.

Steve: VLC... <http://www.videolan.org/vlc/>

Hermann: so my goal to make automated copies...

Steve: If you don't mind sending 4-6GB of data, remake the process, then it's fine.

Hermann: prefer to stick with the original. If we convert we would still keep the ISO. So makes no sense.

Steve: Other questions?

Kate: We've talked about many Q's I had, a little bit about how close captioning and subtitles are carried. Are they tagged or how does that data get stored?

Steve: Subtitles are different kind of tracks so we have audio, video, subtitle.

<http://matroska-org.github.io/matroska-specification/subtitles.html>

You can tag it, explain exactly what tracks are. The author of that track, producer, for just one track. Audio could be different from subtitle, or the same. Authorship, who did the translation, original language. All kinds of stuff like that.

Kate:TimeText?

Steve: Yeah we even support WebVTT. Text-based subtitles either as just basic text with nothing, text with formatting. Or bitmap-based subtitles like in a DVD. The subtitles are in bitmap-format. So there's numerous possibilities of support.

Kate:TimeCode?

Steve: Haha... for me they were the same... so in the spec and everywhere it's written timecode but you should read it as timestamp. Don't yet have timecode support but started the discussion last week of how we can support it in Matroska. No problem in supporting and putting it in the formats but want to know the right way to do it. What I'd like to know is anything you all know -- are they always just the simple text? Or other formats? More information? Can it just be a counter? Turn into a text subtitle track that would just show the text of the timecode that way benefit of using that is it would play in any current mkv player already existing that can play subtitles. If we need a different kind of track... it means nothing will be able to read it presently.

Tobias: Are timecode frame-based?

Steve: No but if they have the same time stamp they will be displayed at the same time. From what I understand, do timecode need to be tied to a frame or for referencing, editing?

Kate: Continuous or discontinuous, it could or could not be, depends.

Steve OK

Discussion about CD text, other things.

Hermann: The use is... if you had 3 camera,s you define 1, 2, 3. Disadvantage is if you forget, you still have the user bit inside. So I don't trust them.

Steve: then it's ancillary data, not time codes. That's not a subject but maybe we can talk about it. One pre-sized use case, format, even if stored in digital bits. It ends up displaying to the user. I'd go with subtitle track if it were me.

Ancillary data that could be analyzed.... Could be coded in a different way. Since it's less important...

Hermann: Importance, is the timecode and the original timecode, the difference.

Steve: diff between time code nad time stamps, timecode is whatever comes from a diff source. We could start from 0 a few times.

Kate: Your time stamp is the same as our master time code. The historical source timecode... EBU R122 <https://tech.ebu.ch/docs/r/r122.pdf> has defined a list which you'd probably want to refer to. Five -- from 122. Two additional time codes in as-07. You'd want just as done in captions want to tag them, label their source. Distinguish between new time stamp and historical time code.

Steve: Yeah they are very different and can't be mixed. Other formats might need to be put in a track.. Probably. The discussion with Dave/Jerome/etc is you want to know when a timecode reset in the file. You want that info so you don't have to play the thing.

Kate: NARA stuff has that, I'd have to think of it as a subtitle track...

Ancillary data.

Steve: We do not store at all and don't know much about them, but what we have in MKV is what I said before, kind of track that is called complex, audio-video-subtitles-control. Anything not in this is complex. Whatever someone wants to put in to the data. Not needed

Ashley: Different form attachments.

Steve: Yes Attachments are beginning or end, in an ancillary data track

Ashley: So it works as a stream.

Steve: Yes.

I don't know how many standards of ancillary data there are but would need ID for complex track, codec ID explaining how to extract data from video how to store it in mkv. Basically mapping from original to how it makes to mkv in and out.

Kate: We carry that data in as-07, another opp to copy that, so we carry that stuff like closed captioning, teletext, SMPTE ST334-1 (Ashley note: tried to link to this but remembered its SMPTE)

Steve: depending on data format, could define either to keep it as original or keep the original minus stuff that's already been extracted to create other proper tracks. But prob for preservation better to keep original.

Kate: We keep them all. It's required in as-07. One thng we have in this is a manifest, xml with a schema for it.

Michael: Is manifest just metadat? Or like bash manifest, etc...

Kate: Just a readable summary of what to expect in the file. These are the parts.

Steve: Basically can tag the segment (the whole mkv file) and put whatever info about that segment in there.

Ashley asks if this manifest describes one asset or could describe many

Kate: Things are broken into parts, Manifest describes each part of the manifest file.

Michael: Ambisonic audio, but any 360 video?

(Michael basically scared room)

Michael: Seeing 360 but vertical is only 45 degrees, or 270 degrees, so much variety we are seeing in terms of recording videos in various dimensions

(conversation around scary dark future of spherical pixels, six walls of a cube, $\frac{3}{4}$ of a sphere, rectangular or other projection).

Michael: Facing these at Youtube so it's useful if a video could be updated and not required to have people update their app on their phone to view it correctly. How can older video players still understand new projection standards?

Meeting concludes because everyone is out of questions!

J/k there are more questions!