Status of CELLAR: Update from an IETF Working Group for Matroska and FFV1

Ashley Blewer
Brooklyn, NY, USA
ashley.blewer@gmail.com

Dave Rice
Brooklyn, NY, USA
dave@dericed.com

ABSTRACT

The open, independent, and international standards organization Internet Engineering Task Force (IETF) has chartered a working group. It is named "Codec Encoding for LossLess Archiving and Realtime transmission" (CELLAR) and aims to develop specifications for a lossless audiovisual file format for use in archival environments and transmission. It consists of the combination of the audiovisual container Matroska, lossless video codec FFV1, and lossless audio codec FLAC. This paper reviews the status of this on-going development and thereby provides an overview of the challenges and intricacies of audiovisual specification development.

Keywords

Format standardization; audiovisual file formats; digital preservation

1. INTRODUCTION

This paper reviews the status of the ongoing work within the Internet Engineering Task Force (IETF)'s *Codec Encoding for LossLess Archiving and Realtime transmission* working group (CELLAR). The working group is tasked with the standardization of three audiovisual formats: Matroska, FFV1, and FLAC. The authors will provide an overview of the challenges, intricacies, and progress of specification development for audiovisual formats. Topics include an overview of the benefits of open standards within the context of digital preservation, methods for advocating for and supporting implementation of standards, and the relationships between specification development and development of validation software.

2. OPEN FORMATS

Matroska, FFV1, and FLAC are open file formats. Their specifications are freely available and openly licensed, continued development is open and available to the public, historical context and conversations surrounding the specification are open access, and use of the formats or their specifications is without charge and can be used by any person. Anyone can improve upon the standards body, contingent only on the standards body to collectively approve of changes.

Matroska as an audiovisual file format has been in use since 2002, with widespread internet usage. Matroska is based upon Extensible Binary Meta Language (a binary equivalent of XML) and is the foundation of Google's webm format -- a file format optimized specifically for web-streaming. Some of Matroska's features -- such as subtitle management, chaptering, extensible structured metadata, file attachments, and broad support of audiovisual encodings -- have facilitated its adoption in a number of media communities. Matroska has also been implemented into many home media environments such as Xbox and Playstation and works "out of the box" in the Windows 10 operating system.

The Matroska wrapper is organized into top-level sectional elements for the storage of attachments, chapter information, metadata and tags, indexes, track descriptions, and encoding audiovisual data. Each element may have a dedicated checksum associated with it, which is one of the important reasons why it is deemed such a suitable format for digital preservation. With embedded checksums, a specific section of a Matroska file can be checked for errors independently, which means error detection can be more specific to the error's region (as opposed to having to identify the error within the entire file). For example, a checksum mismatch specific to the descriptive metadata section of the file can be assessed and corrected without requiring to do quality control and analysis on the file's content streams. The Matroska format features embeddable technical and descriptive metadata so that contextual information about the file can be embedded within the file itself, not just provided alongside in a different type of document.

FFV1 is an efficient, lossless video encoding that is designed in a manner responsive to the requirements of digital preservation. FFV1 has rapid traction in both the development and digital preservation communities and is widely and freely distributed with the ubiquitous ffmpeg and libav libraries for video processing. FFV1's lossless compression algorithm allows uncompressed video to be reduced in filesize without loss of quality while adding self-description, fixity, and error resilience mechanisms. FFV1 version 3 is a very flexible codec, allowing adjustments to the encoding process based on different priorities such as size efficiency, data resilience, or encoding speed. FFV1 is a strong candidate for video files undergoing file format normalization prior to the OAIS-compliant repository ingestion phase. For example Artefactual's Archivematica (a free and open-source digital preservation system) uses FFV1 and Matroska as a default normalization strategy for acquired audiovisual content and recommends pre- and post-normalization FFV1+MKV validation methods [4] [8].

FLAC is a lossless audio codec that has seen widespread adoption in a number of different applications. FLAC features embedded CRC checksums per audio frame, but also contains an md5 checksum of the audio stream should decode to. Another benefit of FLAC is that it can store non-audio chunks of data embedded in the source WAVE file, such as descriptive metadata. Since FLAC is designed to store foreign data (using the --keep-foreign-metadata option), it is feasible to encode a valid WAV file to FLAC (which adds several fixity features while reducing size) and then extract the FLAC back to recreate the original WAV file bit for bit. Tools such as the flac utility and ffmpeg can analyze a FLAC file to identify and locate any digital corruption through the use of the format's internal fixity features.

3. SPECIFICATION-BASED VALIDATION

Developers of open source software have been building tools based on the Matroska specification for many years. MKVToolNix is a suite of software tools created to work specifically with Matroska files designed by Moritz Bunkus, a core developer of Matroska and EBML. A part of mkvtoolnix is mkvalidator, which is described as "a simple command line tool to verify Matroska and WebM files for spec conformance" [3]. To facilitate that the specification is well-interpreted by the developers of tools that implement it, the mkvalidator tool provides a programmatic assessment of the validity of a Matroska implementation, whereas the specification itself is meant for a human reader. The co-development of an official specification and an official validator provides a means for both humans and computers to assess and interpret the quality of a Matroska deployment. This co-development of the specification and validation tools should be considered as a model in the specification of other file formats as well.

MediaConch is software currently being developed as part of the PREFORMA project, co-funded by the European Commission. The PREFORMA consortium describes the goal "is to give memory institutions full control of the process of the conformity tests of files to be ingested into archives" [7]. The goal of the PREFORMA project is to create open source software for the most eminent archival-grade media formats: PDF, TIFF, Matroska, FFV1 video, and LPCM audio (with MediaConch focusing on the latter three). These software packages focus on the validation and conformance checking of files against their official specifications. Investigation into the development of this software has sparked conversations on the related format list-servs (Matroska-devel, ffmpeg-devel, and libav-devel) and in other public platforms like GitHub. This investigator and conservation helped raise awareness of the state of the existing specification documents and need for more format and structure standardization processes through an established open standards organization. With a collaboration between related developer and archival user communications a proposal for a working group focused on lossless audiovisual formats was submitted for the consideration of the IETF, which would become the cellar working group.

The MediaArea team (developers of MediaConch) has been working on understanding the specific details of each segment of an archival video standard, sometimes down to the bit-level, in order to develop a comprehensive conformance checker. MediaArea has previously developed Mediainfo, a command-line software application prolifically used in media archives to quickly assess file information, and MediaTrace, developed with MoMA to provide bit-level analysis on media files.

4. EARLY STANDARDIZATION WORK

Matroska and EBML were developed from the beginning with standardization in mind. The conceptual file formats, the documentation, and associated software and libraries were developed and implemented simultaneously by the same core team. The authors of Matroska documentation were also developing validation tools such as mkvalidator, so that there was both a human-readable and programmatic methods to test if a produced Matroska file adhered to the specification or not. With other file formats, the specification and validation tools are generally developed separately by distinct teams. As lead contributors to Matroska's core libraries and validation tools are written by the same authors of the specification, there is an opportunity for the interpretation of the specification to be very clear and precise.

Matroska's history contained many pushes in the direction of more official standardization. In 2004 (two years after the origin of Matroska), Martin Nilsson produced an RFC draft of EBML, which extensively documented the format in Augmented Backus-Naur Form (ABNF) [6]. This draft was not published by the IETF but remained on the Matroska site as supporting documentation. Also in 2004, Dean Scarff provided draft documentation for a concept of the EBML Schema. An EBML Schema would be analogous to the XML Schema for EBML Documents and could provide a standardized structure to define EBML Document Types such as Matroska and webm. Additionally extending feature support and clarifications to documentation would be ongoing themes of the development listserv.

FFV1 was initially designed and incorporated into FFmpeg in 2003 as an experimental codec. Early documentation may be seen in the Internet Archive [5]. In 2006, FFV1 was marked as stable and gained use as a lossless intermediate codec to allow video to be processed and saved to a file without impactful encoding loss or the large sizes of uncompressed video. Between 2006 and 2010 FFV1 performed favorably in lossless video codec comparisons and found some early adoption in archives. However, at the time FFV1 had notable disadvantages compared to other lossless encodings used in preservation such as JPEG2000 and Lagarith, including a lack of support for 10 bit video, need for optimization, and crucially-needed documentation and standardization efforts.

From 2010 through 2015 FFV1 underwent significant developments and increased archival integration. Michael Niedermayer, the lead format developer, significantly expanded the documentation and released FFV1 version 3, which added embedded checksums, self-description features, improved speeds with multi-threading, error resilience features, and other features that improves the efficiency of the encoding in preservation contexts. Kieran Kuhnya, Georg Lippitsch, Luca Barbato, Vittorio Giovara, Paul Mahol, Carl Eugen Hoyos and many others contributed to the development and optimization of FFV1. In 2012, work on the specification moved to more collaborative environments in a GitHub repository. During this time, archival experimentation and implementation with FFV1 expanded and many archivists (including the authors of this paper) actively participated in supporting the testing and development of FFV1's codec and documentations.

Michael Niedermayer began documenting a specification for the format and added several features specific to preservation usage. Version 3 is highly self-descriptive and stores its own information regarding field dominance, aspect ratio, and color space so that it is not reliant on a container format alone to store this information. Other streams that rely heavily on their container for technical description often face interoperability challenges.

Much like Matroska, despite the widespread usage, the FLAC file format had not been through a process of standardization in a standards body. However the FLAC development community has authored and maintains a comprehensive specification on the FLAC website.

5. STANDARDIZATION

The IETF

The Internet Engineering Task Force (IETF) is an open and independent international standards organization, known for the development of standards for the Internet protocol suite (TCP/IP), file transfer protocol (FTP), and protocols that

compose the Simple Mail Transfer Protocol (SMTP). IETF's parent organization is the Internet Society (ISOC), an international, non-profit organization that has set out to "make the world a better place" by "connecting the world, working with others, and advocating for equal access to the Internet" [2]. Much of the standardization work shepherded by IETF focuses on the development of standards of and is related to the transmission of information between systems in an efficient manner without error or data loss.

The working methods of IETF promote and ensure a high degree of transparency so that anyone is able to look upon processes underway and to participate within them. Communication is organized into a system of publicly accessible mailing lists, document trackers, and chatrooms. The IETF's conferences (held three times per year) include audiovisual streams, IRC streams, and an in-room facilitator for remote participants to efficiently invite and enable participants in the process.

PREFORMA

The PREFORMA Project is a Pre-Commercial Procurement (PCP) project started in 2014 and co-funded by the European Commission under its FP7-ICT Programme. The project responds to the challenge of implementing good quality standardised file formats within preservation environments with a particular focus on providing memory institutions with control over conformance and validation testing of those file formats. Along with PDF and TIFF, the PREFORMA administrators selected Matroska, FFV1, and LPCM as open file formats of particular interest to preservation communities and selected MediaArea to develop conformance tools for those formats.

In early planning, MediaArea's team (including the authors of this paper) noted the particular challenges in developing conformance tools for file formats whose specifications had not yet been subject to the procedures and protocols of a standards body. PREFORMA's network of developers and memory institutions provided an environment supportive of collaboration between developers, specification authors, and archivists. Format maintainers, developers, and archivists collaborated to participate and encourage work on Matroska and FFV1 within an open and inclusive standards organization.

The IETF as a Standards Body for Audiovisual Preservation?

Through consensus with participating communities and public discussion, the IETF was selected as the most suitable standards body with which to standardize FFV1 and Matroska due in part to its open nature, transparent standardization process, facilitation of accessibility, and organizational credibility. IETF lacks paywalls and licensing barriers for accomplished and published works. IETF provides ability for all interested persons (members or not) to participate via multiple open channels. Additionally the related developer communities of ffmpeg-devel, libav-devel, and matroska-devel were well familiar with IETF either from involvement in earlier standardization efforts and IETF's expanding role in standardizing audiovisual formats, such as OGG, VP8, and Opus.

Participants from Matroska, FFmpeg, PREFORMA, MediaArea and many other communities collaborated to propose the formation of an IETF working group to standardize lossless audiovisual file formats for preservation. Tessa Fallon presented a draft charter at the dispatch working group meeting at IETF93. The IETF approved the charter for the working group, named CELLAR (Codec Encoding for LossLess Archiving and Realtime transmission). The opening sentences of the CELLAR charter read as follows: "The preservation of audiovisual materials faces challenges from technological obsolescence, analog media deterioration, and the use of proprietary formats that lack formal open standards. While obsolescence and material degradation are widely addressed, the standardization of open, transparent, self-descriptive, lossless formats remains an important mission to be undertaken by the open source community" [1]. CELLAR's goal is stated as being "to develop an official internet standard for Matroska (audiovisual container), FFV1 (lossless video encoding), and FLAC (lossless audio encoding) for use in archival environments and transmission" [1]. This process involves the further testing and development of the specifications of these three formats to ensure their sturdiness, success, consensus, and maintenance long into the future.

CELLAR Happenings

The work of the CELLAR Working Group can be seen, commented upon, or contributed to in a few working spaces. The mailing list is the central location for communication and discussion on works towards the working group's objectives. The mailing list, along with other central information pertaining to the working group, is located at: https://datatracker.ietf.org/wg/cellar/charter/
The mailing list archive is available at: https://mailarchive.ietf.org/arch/search/?email_list=cellar
At the time of publication submission on 17 July 2016, the mailing list of the working group includes the participation of 82 individuals.

For both Matroska and FFV1, the working group is building upon earlier specification work done independently by the formats' designers and contributors. A first important step in the process was making the specifications more accessible by improving their online presence. Both Matroska and FFmpeg managed in-development specification drafts on their websites with contributions from the community. Within the IETF working group this development continues with improvements to the specifications themselves and improvements to the websites that support those specifications with the goal of allowing more collaborative work by an expanded population of developers and archivists. The FFmpeg specification webpage was formerly built in LyX. In Summer 2015, the specification was migrated to Markdown, a syntax easier to read and easily hosted on collaborative version control platform, Github. Similarly, the Matroska specification was hosted in the main Matroska website, built in Drupal. It has also been migrated to Markdown and Github to promote collaboration of specification refinement work done primarily in conversation via the CELLAR listserv.

Accomplishments via CELLAR

CELLAR work has resulted in producing valid RFCs for EBML, Matroska, and FFV1 for official consideration at IETF's July 2016 conference. These RFCs are early draft specifications constructed through restructuring, clarifying, and building upon the existing specification as well as adding sections mandated by RFC guidelines such as security considerations, abstracts, and references.

Overall the work of cellar has fallen into three categories. 1) Meeting IETF's documentation requirements through adding mandated sections such as security considerations, valid references, abstracts, and notations. 2) Improving existing documentation, such as rewriting and refining what has already been put into practice but needs fine-tuning. 3) Extending

features to accommodate new use cases and respond to past lessors learned.

New features have been proposed and added to the updated specification, including a proposal for color management (via Google in relation to WebM), disambiguation and refined specification for timecode, and improvements to interlacement status.

Existing features require further clarification, and much work has been done in this area. This involves gather use cases, reviewing the existing specification, and fixing discrepancies between elements and clarifying the language when vague or able to be interpreted (or have been interpreted) in different ways.

Within the working group the sections of Matroska's specification that pertained to its underlying EBML format where consolidated into a EBML specification, so that the Matroska specification may build upon the EBML specification rather than act redundantly to it. The updated EBML specification includes documentation on how to define an EBML Schema which is a set of Elements with their definitions and structural requirements rendered in XML form. Matroska's documentation now defines Matroska through an EBML Schema as a type of EBML expression.

RFC drafts have been submitted in anticipation of IETF96 and the CELLAR working group meeting (held on 19 July 2016). During this meeting, the specification will be reviewed. Comments will then be discussed and implemented into the next version of the EBML RFC. There is still a long way to go in refining these RFC documents to IETF standards and consensus as can be seen in the comprehensive reviews arriving at the cellar listserv prior to the working group meeting.

The work of the cellar working group is ongoing and active. The working group provides a unique environment where archivists are working alongside developers and specification authors.

6. FORMAT RECOMMENDATIONS

The specifications of Matroska and FFV1 permit a range of flexible usage to accommodate distinct use cases and priorities. Specific uses certainly benefit from specification optimization and recommended practice. Best practices for the usage of both Matroska and FFV1 are evolving due to the work of the CELLAR working group. However the authors of this paper would like to present recommendations for optimization for current use and look to what may be useful in future refinements of FFV1 and Matroska intended specifically for digital preservation.

The benefits and security of whole-file checksums do not scale fairly for larger audiovisual files. Whereas an electronic records collection may store thousands of files in the space of a terabyte and thus manage thousands of corresponding checksums to authenticate the storage, an audiovisual collection may use a terabyte to occupy a few dozen files. The larger the file is, the less effective a checksum mismatch is at clarifying the extent and location of the error. Both FFV1 and Matroska incorporate fixity features so that pieces of the data utilize their own checksums.

Matroska adopts of feature of its foundational format EBML, which supports nested checksum elements into any structural element container. The EBML specification states "All Top-Level Elements of an EBML Document SHOULD include a CRC-32 Element as a Child Element." This enables attachments, track metadata, description metadata, audiovisual data and all other sections to have the ability to manage their own checksum. This allows a much more granular and targeted use of checksums and also enables parts of the file to be changed while maintaining the fixity of the other parts. For instance a Matroska file may store audiovisual content, attached images of the source video tape, and logs of the creation of the file. Add a later stage in the archival life of the Matroska file, a quality control report may be created about the file and then itself stored within the file without affected the fixity of the audiovisual data.

FFV1 version 3 mandates the storage of checksums within each frame so that the decoder may know precisely if a frame is valid or invalid. Optionally FFV1 version 3 can incorporate checksums into each slices of the frame. In this case, if the data is corrupted the decoder can know what region of the frame is damaged and conceal it by duplicating pixels from the previous valid frame into the corrupted space. FFV1 is able to re-use contextual information from frame to frame as a way of reducing its data rate; however the re-use of context across frames can reduce the error resilience of FFV1. In preservation it is recommended that all FFV1 frames are encoded as self-dependent so that they are not dependent on information from another field. This is done by setting the GOP (group of pictures) size of the FFV1 encoding to 1.

FFV1 encodings are generally much faster than other lossless encodings partly because of the support of multithreaded encoding. With multithreaded encoding the frame is sliced into many slices that are encoded through separate processes and merged back into a frame. Encoding with slices also reduces the visual effects of data corruption by regionalizing damage to a smaller contained area. It is recommended to use a higher slice count such as 24 or 30 while encoding to benefit from these features.

FFV1 version 3 incorporates significant preservation features over the prior versions. Within version 3, the micro versions of 3.1, 3.2, and 3.3 were experimental and version 3.4 was the first stable release. So specifically, version 3.4 is recommended.

Both FFV1 and Matroska incorporate a significant amount of self-description. It is recommended that such metadata be declared specifically (rather than noting an 'undetermined' value) and that the metadata is consistent between the Matroska container and FFV1 encoding. For instance FFV1's picture_structure value should clarify the interlacement and not be set to 'undetermined' unless it truly is undetermined. Additionally FFV1's sar_num and sar_den (which document sample aspect ratio) should be explicitly set rather than set as '0' which would indicate an unknown sample aspect ratio.

As videotapes are digitized there is a lot of contextual information to clarify. Videotape players generally do not communicate values such as audio channel arrangement or aspect ratio (especially true with analog media). A videotape may have traditional default considerations, such as considering the first audio channel as left, second as right, and aspect ratio as 4/3; however, this should be clarified in the digitization process and not left to assumption. It is recommended that values such as aspect ratio and audio channel arrangement be set explicitly where possible.

Often a physical audiovisual carrier is not able to communicate the aperature or boundary of the image during digitization. For instance a 720x486 encoding of video may only contain a 704x480 active picture bordered by rows and columns of black pixels. Alternatively a film scan may include film perforations, sound track data, or the border between frames. The framing of the presentation can be clarified using Matroska's PixelCrop

elements. This allows the active picture to be set according to coordinates while preservation the entirety of the encoding image. This feature can also allow black pixels from letterboxing or pillarboxing to be hidden or possibly to hide head switching or unintended video underscan from the presentation while preserving it.

Legacy videotape does not contain a method for a machine to understand where the content starts and ends. Additionally legacy videotape often contains supporting content for technical and historical reasons. For instance a 30 minute program on videotape may be included with several other minutes of color bars, informational slates, countdown, black frames, and other material not intended to be part of the presentation.

Matroska's chaptering support includes a feature called Ordered Chapters. With Ordered Chapters a user be document various intended presentations of the video. For instance, one Matroska file may contain a set of chapters that presents the entirety of a digitized videotape (including color bars, static, black frames and whatever else is present). The same file may contain another edition of chapters that presents only the featured content of the tape and skips over the colorbars and other technical video content. Players such as VLC provide means to switch between chapter-managed presentations. It is recommended to showcase the intended presentation with the default edition of chapters and provide access to the full encoding of the videotape's content via an alternate edition of chapters.

Matroska has a strong focus on managing language for subtitles, audio, and metadata. While Matroska defaults to English, it is recommended to clarify language properly, so that if a file contains many alternate audio encodings or sets of metadata that their language is properly marked.

Recommendation Summary (ffmpeg options are in backticks):

When storing content in Matroska for preservation use CRC-32 Elements in all Top-Level Elements as suggested by the EBML specification.

When encoding FFV1 for preservation include the options: `-level 3` and `-slicecrc 1` to request FFV1 version 3 with slice crcs enabled.

Use an FFV1 GOP size of 1 with `-g 1`.

Use a high slice count (at least 24) during FFV1 encoding, `-slices 24`.

Avoid setting FFV1 values of picture_structure, sar_num, sar_den to an 'unknown' value.

Use of FFV1 of at least version 3.4 (major version 3 and micro version 4).

Be as explicate and accurate as possible when storing information about aspect ratio, audio channel arrangement, presentation timeline, and language.

Consider using Order Chapters to distinguish the intended presentation of a digitized videotape from other technical content (such as color bars and countdown).

Also of these recommendations are feasible with mkclean, mkvpropedit, and ffmpeg or avconv.

7. CONCLUSIONS

Historically, many digital audiovisual formats put forth as archival standards have been proprietary and have fallen out of common usage as they become outdated, unpopular, or as the governing company loses interest in seeing the project continue. The specifications of both FFV1 and Matroska have been actively developed in an open source and open license environment that welcomes participation and review. Many of the prominent contributors and authors of these specifications also concurrently contribute to the development of open source tools to utilize, assess, or integrate these formats. As a result, the specification development isn't wholly idealistic but the design effort is tied to ongoing contributions to the main open source projects that support the specifications. The work in CELLAR to improve the specifications is an effort that parallels efforts in VLC, Libav, FFmpeg, MKVToolNix, and other open source toolsets that deploy the new aspects of the specification.

FFV1 has been at a tipping point in adoption within the preservation community. Archivematica has adopted FFV1 for lossless video normalization for long term preservation. More digitization vendors have added support for the format as well. Matroska has been under a slower adoption by archives but its features for sectional fixity, hierarchical metadata, attachments, and preservation data make it worthy for consideration. Additionally as the specification is open source and its refinement is in an active IETF working group that specifically focuses on archival use, archivists are encouraged to review and participate in this effort.

8. REFERENCES

[1] IETF Cellar Working Group, "CELLAR Charter," [Online] Available: https://datatracker.ietf.org/wg/cellar/charter/

[2] Internet Society, "Internet Society Misison," [Online] Available: http://www.internetsociety.org/who-we-are/mission/

[3] Lhomme, Steve and Bunkus, M., "mkvalidator tool," [Online]. Available: https://matroska.org/downloads/mkvalidator.html

[4] McLellan, Evelyn, "Format Policies," [Online] Available: https://wiki.archivematica.org/Format_policies

[5] Niedermayer, Michael, "Description of the FFV1 Codec," [Online] Available: https://web.archive.org/web/20030807095617/http://mplayerhq.hu/~michael/ffv1.html

[6] Nillson, Martin, "EBML RFC (Draft)," [Online] Available: https://matroska.org/technical/specs/rfc/index.html

[7] PREFORMA Consortium, "PREFORMA Future Memory Standards," [Online] Available: http://preforma-project.eu/

[8] Romkey, Sarah, "Requirements/MediaConch integration," [Online] Available: https://wiki.archivematica.org/Requirements/MediaConch_integration

9. CREDITS

Many thanks to Natalie Cadranel, Sebastian G., Kieran O'Leary, and Erwin Verbruggen for their contributions and suggestions to this article.