



Mastering the Integrity of Technical Information in Tapeless Production and Long-Term Archives

By Benoit Fevrier, Valérie Popie, and Ludovic Dupont

The main challenge for information technology teams is to ensure that the media created today will not become obsolete tomorrow. Standards must remain high, while all types of information continue to be accessible. In workflow architecture, key decisions revolve around choosing the right audio/video codecs to guarantee maximum interoperability. However, although various layers such as vertical ancillary (VANC), vertical blanking interval (VBI), SMPTE 328M, and video auxiliary (VAUX) contain strategic data (time codes, multichannel audio descriptions, closed captioning, aspect ratio, etc.), they are not sufficiently taken into account in operation processes. This valuable information often comes embedded in the input source video stream and must be directed toward the Material Exchange Format (MXF) essences to ensure a genuine integrity of metadata in a multilayered bark. This document provides practical case studies on codec selection, time code and subtitle carriage, as well as the number of audio channels, including those encoded in Dolby-E.

INTRODUCTION

This document does not aim to explore all engineering topics referred to in depth, but rather to focus on the interoperability issues that arise during audiovisual (AV) workflow implementation.

Because of the numerous wrappers that envelop tapeless files, each layer has labels carrying specific information. When a file is unwrapped to exploit its audio and video content, different systems may require the removal of one or several wrappers, depending on the workflow and the target. The question that then comes up concerns the integrity of the data, particularly when they are redundant with different values: is the data indeed valid, and can it be used for the workflow operations?

When broadcasters move to a high-definition (HD) tapeless workflow, they have usually worked out its video codec and its associated properties description. However, although other data such as audio or subtitles seem less important, they nonetheless represent a significant part of the AV content. Mastering all properties that exist in the different layers of an AV file is essential not only for daily production but also for long-term archiving. Poor information management can lead to a lack of interoperability and should

be approached from a more general information technology viewpoint. In AV asset preservation, the challenge is far more complicated. The AV files must have a maximum of valid data to be usable in the short, middle, and very long term.

COMPOSITION OF AN AV PROGRAM

Let's start with a simple question: "What is an AV program made of?" The answer is in the question: audio and video content. Those who have worked in the broadcast industry over the past decades know that an AV program needs additional information to ensure its integrity. The ancillary data that accompanies AV content in video transmission or on videotape existed long before the term metadata was invented. That said, basic ancillary data is required to make an AV program operational in the broadcast industry:

- The first that comes to mind is time code information. A video program with no time code information would be inconceivable in the broadcast industry.
- An AV program is often complemented with subtitles, whether they are closed caption or Teletext.
- Aspect ratio in production or archive is important. The video standard always sets the size of the frame, with its number of lines and number of points in a line, but the pixels in a file are not always square. Still, a correct aspect ratio is key if the broadcaster wants a program to have a "normal" proportion.
- Audio is now multichannel for multilanguage audio, spatial 5.1, or audio description.

We now consider AV content in video and audio tracks, time code and subtitle data, and other essential information, such as aspect ratio and multichannel audio.

SCHEMATIC DIAGRAM OF AN AV FILE

Today, an AV program is made up not of a physical device but of an immaterial file. **Figure 1** presents a schematic view of an AV file.

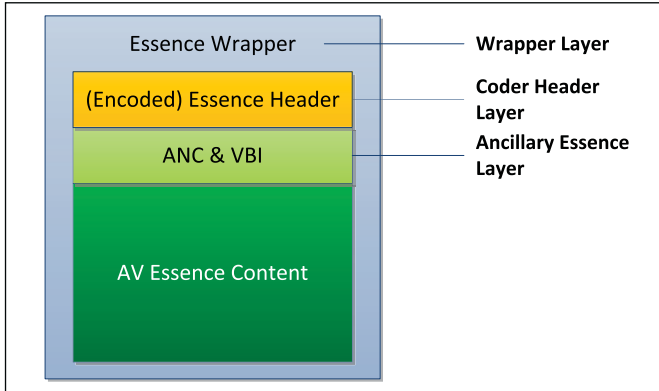


Figure 1. Different layers carrying ancillary data and metadata in an AV file.

In this diagram, three layers can be defined:

- First layer—ancillary essence layer: A layer composed of vertical blanking interval (VBI), ancillary (ANC) or Audio Engineering Society (AES) data. It is embedded in a baseband Serial Digital Interface (SDI) video signal.
- Second layer—codec header layer: When the AV content is encoded, the related codec always generates a header that stores auxiliary information. In this document, we focus mainly on MPEG-2 codec headers.
- Third layer—wrapper layer: This layer envelops all data necessary to make an AV file. There are many types of wrapper formats, more or less standard and more or less well supported. In this document, we focus on the Material Exchange Format (MXF) standard.

MANAGING THE INTEGRITY OF TECHNICAL INFORMATION

In the following sections, we describe some interoperability issues encountered in “real-life workflows”—issues that stem from technical data that is stored in multiple locations in the multiple layers of an AV file.

Managing Multilayer Time Code Data

European Broadcasting Union (EBU) Recommendation R122 provides a comprehensive list of the many locations where the broadcaster can find time code data in an MXF AV file.¹ We determine the interoperability issues that can arise from a redundancy of information in the different layers described previously.

When AV content is digitized from baseband SDI, the video stream carries the vertical interval time code (VITC) in the ancillary essence layer (VBI or VANC data). In a standard definition workflow, which is often based on the D-10 format,² the number of active lines per frame is 608 lines in 625/50 systems and 512 lines in 525/60 systems. When D-10 is chosen as a production or archive standard, the AV file transports the VITC information in the VBI layer; this information can be used as required during the media life cycle. Several cases occur in real life:

- In production, when broadcasters ingest a tape and generate a file, they may want to reassign a time code to the program (e.g., in the wrapper layer). This new time code value is changed in the wrapper and/or in the media asset management but often remains in the VITC/VBI with the original, and thus false, value.
- In archive production, preserving the time code information can be essential, because it is often linked to a time code–based archive asset management cataloging. A tape may have hosted several programs and as a result have a discontinuous time code. When the tape is digitized to a file, the broadcaster often wants to clean the program by regenerating a continuous time code. This new time code is frequently handled in the wrapper layer but sometimes is left by mistake as VITC/VBI (**Fig. 1**).
- In post-production, when the finalized program is issued from a composition, the VITC can be blanked at times by the editing system.

Broadcasters must then deal with these problems. Poor management can lead to unfortunate consequences, especially for the final stage playout. On the other hand, broadcasters want their video asset to have maximum information integrity. Archiving files with the D-10 standard, which is now widespread, can be risky if the VITC/VBI information is incorrect or missing, especially when no one can predict which system will be managing the information in 20 or 30 years.

The codec header layer has also a lot of “cabins” to host time code data. Let’s take one example and consider the MPEG-2 codec that is frequently used in both standard-definition (SD) and HD workflows. The MPEG-2 ISO/IEC 13 818-2 norm allows the broadcaster to put the time code in different locations, notably in the MPEG-2 group-of-pictures (GOP) header structure.³ The time code can also be nested in the MPEG-2 “user data” carriage, as described in the SMPTE 328M standard.⁴ When an AV file is encoded in MPEG-2 and happens to host user data—which occurs frequently and concerns more than just the time code—broadcasters have a minimum of two locations to worry about when it comes to pinpointing the right time code information.

As for the time code carriage in the wrapper layer, the MXF standard has powerful tools that allow users to set the time code in different locations in the wrapper structure, including the following:

- A time code track in the material package (MP) (in the SMPTE ST377-1 MXF standard,⁵ it is optional, but assuming it is the “playback” time code, it is highly recommended that the MXF file have one).
- Zero or several time code tracks in the (top level) file package or packages (FPs).
- A system item that hosts the SMPTE ST12-1 time code,⁶ as described in SMPTE 405M⁷ and in SMPTE ST331.⁸

The MXF SMPTE standard documents describe, in detail, the purpose of each time code in each location; however, there is always a gap between theory and reality. The result is that there are a multitude of solutions on the market, each of them reading the time code in different locations—not to mention those incongru-

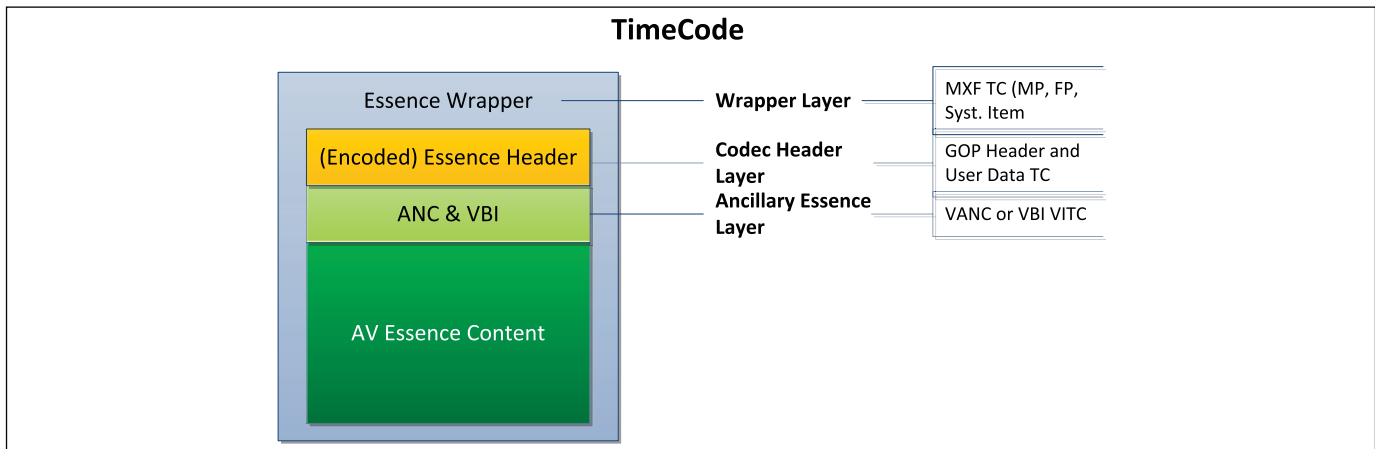


Figure 2. Locations of time code in different layers.

ous MXF generators or readers that use time code information in “nonstandard” MXF metadata. The most logical thing to do would be to set the same value in each time code location, but this rarely happens in real life. **Figure 2** shows the different locations that time code information may have in each layer.

We have shown in brief that even within a layer (codec or wrapper), the time code has a certain talent for ubiquity. And assuming that each time code value has its own integrity, there are simply too many choices, so equipment often reads the time code in the wrong locations.

However, a solution to this problem does exist. You can, for example, set the same time code value in each carriage, and perform a “check-up” on each one whenever an AV file is modified.

Managing Multilayer Subtitle Data

We have often experienced workflow implementations in which an AV file is supposed to carry subtitles but does not output them when played back. Problems with subtitle information is not a consequence of too many carriages in each of the layers, as we have just seen with time codes, but of the process depending on whether the video format is SD or HD and on the type of encoder that was used to generate the AV file.

Subtitle data—whether Teletext or closed caption—is transported in the baseband SDI video signal to the ancillary essence layer before entering the file-based workflow. When the content is in SD format, subtitles are usually transported in VBI inactive lines. In HD, they are often carried in the ANC data. As we explained earlier, VBI lines may remain in a file, especially in the case of the D-10 format, which is commonly used in SD production and archive preservation.

The codec header layer is also the carriage used to store and transport subtitles. We have often handled AV files, particularly in MXF MPEG-2, where closed captioning was located in MPEG-2 user data in the Advanced Television Systems Committee (ATSC) A53/4 format⁹ or some Teletext was just shaped in a RAW ancillary packet in SMPTE 328M.⁴ Experience shows that some playout servers support closed captioning when it is located in this carriage.

Thus, VBI inactive lines are commonplace for subtitles in SD, and coder headers are common for HD. The MXF wrapper layer had a bridge to gap. This is achieved in the MXF standard with the SMPTE 436M track,¹⁰ which is used to transport ANC and VBI information to a separate MXF data track and which provides an easier way to transport and access ancillary video data within the wrapper layer.

Some operations were carried out in SMPTE to specify nonessence data carriage within an MXF generic stream partition, as referred to in SMPTE 410M.¹¹ When this method is adopted by the community of MXF vendors, it will have the advantage of allowing users to format the subtitle data before wrapping it. A complex data model exists to describe subtitle data such as Spruce Subtitle File (STL)¹² and Distribution Format Exchange Profile (DFXP), which is based on World Wide Web Consortium (W3C) Timed Text Markup Language.¹³ Transporting the subtitle data into a nonessence track will certainly prove to be worthwhile, but for the moment it remains purely theoretical. The digital cinema industry started pragmatically, with Extensible Markup Language (XML) subtitle files near the MXF digital cinema package (DCP) files; they have now adopted timed text implementation in MXF to carry and exploit subtitles for digital cinema playout, as defined in SMPTE ST429-5.¹⁴ **Figure 3** provides a concrete illustration.

Contrary to the time code, the real problems with subtitles are not because of redundancy but rather because of forthcoming standards. In the production environment, subtitles are often embedded in the baseband SDI video signal and have a different life cycle in the form of an EBU STL file¹² which is reconciled with the content at the playout stage. In tape-to-file archive migration, the problem still exists: leaving the subtitle in the video signal may not be sustainable in the long term, because relying on a video codec header makes the broadcaster dependent on the codec technology. Despite numerous initiatives, the MXF wrapper is only just ready to propose a viable solution for timed text; as a result, many broadcasters have turned to MXF SMPTE 436M to convey ANC/subtitles data in an MXF track. However, subtitles can be effectively managed in AV tapeless production if the subtitle management process has been fully integrated in the workflow design.

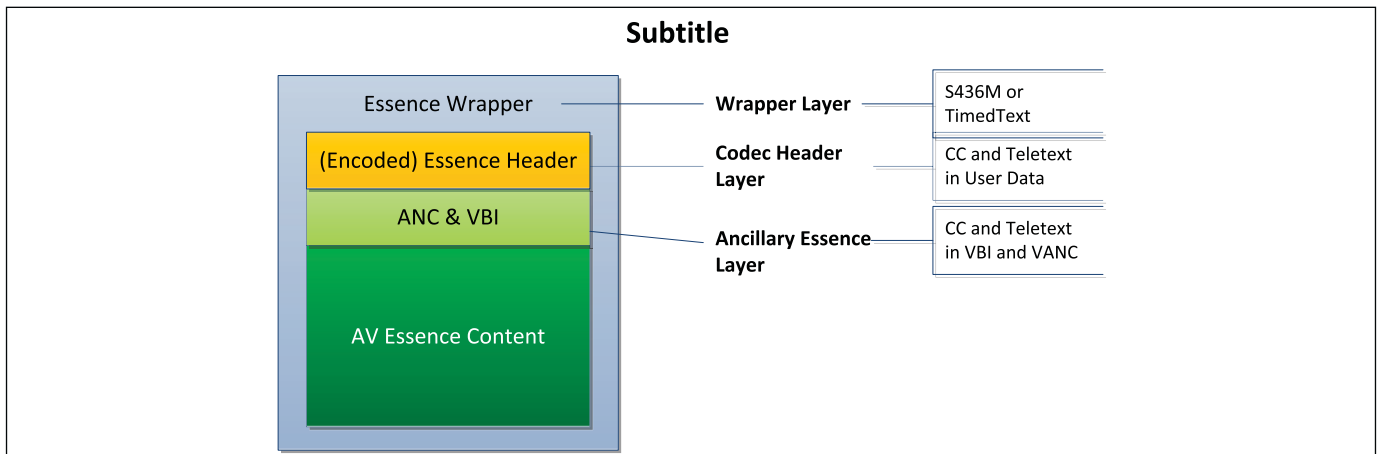


Figure 3. Locations of subtitles in different layers.

Managing Multilayer Aspect Ratio Data

Aspect ratio has a lot in common with time code, because the volume of data is small but its value is of great importance for an AV workflow. Again, the information is there from the start when the content is embedded in the baseband SDI video. The data is then propagated to all layers of the AV file with a varying level of integrity preservation.

Aspect ratio is important information for SD content. The ancillary essence layer, in particular, carries the widescreen signaling (WSS)¹⁵ in line 23 in the phase-alternation line color system (PAL) and is commonly used in Europe. However, even if the AV content is made of active lines, the first active video line is the first half of the 23rd line, while the WSS information is located in the other half line, so it is often ingested and kept as is in the active video of the AV file. In HD production, we sometimes see the active format description (AFD) carried in the ancillary data of HD-SDI video signals in accordance with SMPTE ST2016-1.¹⁶

Many codec header layers also have places to store the aspect ratio information. This is the case for the MPEG-2 video elementary stream, for example, which has a location for the aspect ratio in the MPEG-2 sequence header.³ The aspect ratio data located in the codec header

layer is used as valid information by many systems in file-based workflows. Therefore, encoded MPEG-2 HD or SD transports at minimum two separate aspect ratio data instances. If by chance the aspect ratio is made up of WSS in the VBI and of AFD in the MPEG-2 header, each of them may have a different value with a different meaning.

As for the wrapper layer, MXF SMPTE standards have metadata in which to store the aspect ratio. In the picture essence descriptor, there is an aspect ratio property (“best effort”) and an AFD property (“optional”). The result is that in the real world we often find MXF files without any aspect ratio data in the MXF metadata layer. **Figure 4** gives an overview of the different locations the aspect ratio may have in each layer.

When there is aspect ratio data in the different layers of an MXF file, experience has shown that the value is not necessarily valid. Although in HD this rarely happens, it can be a headache in SD production workflows—where the information is valuable for the upconverting process, for instance—and in video archive preservation when the aspect ratio information is valid only in the media asset management but is incoherent in the AV file layers. There is a definite risk that after several decades no one will remember which information is correct. However, like the time code, it is possible

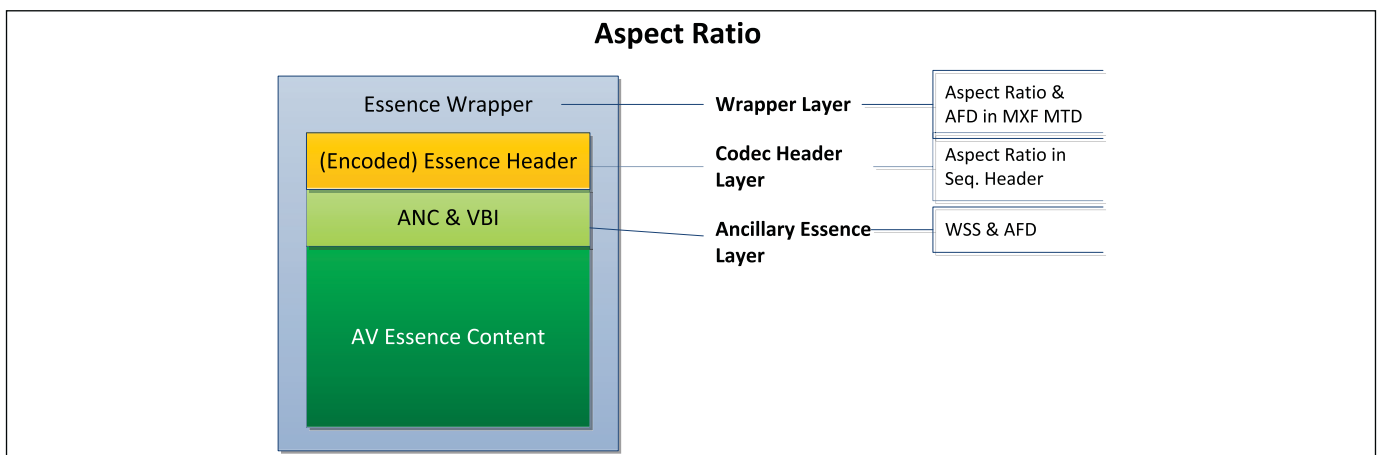


Figure 4. Locations of aspect ratio in different layers.

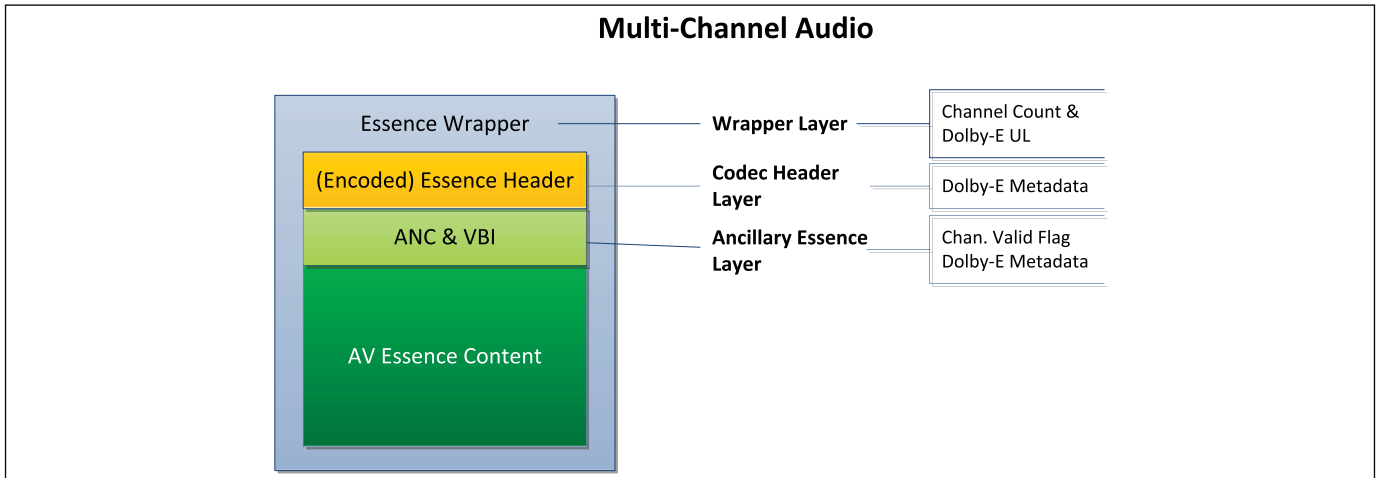


Figure 5. Locations of multichannel audio information in different layers.

to manage aspect ratio efficiently if you have a complete list of the locations of aspect ratio data within the AV file and ensure that each one preserves its data, even after the file has been modified.

Managing Multichannel Audio Data

The audio fragment of an AV file is often multichannel. This multichannel audio stream can be either uncompressed or compressed. Let's work out how this multichannel audio data information fits into a multilayer AV file.

The multichannel audio streams are transported in the baseband SDI as ancillary data shaped in the AES3 stream as described in SMPTE 272M¹⁷ for SDI and in SMPTE ST299-1¹⁸ for HD-SDI. When this audio stream is ingested from the baseband SDI and wrapped in an AV file, the AES stream is often stored as is in the file. We already mentioned that D-10 wrapped in MXF is a popular format for SD file-based production and archive workflows. When wrapped in an MXF D-10 generic container,¹⁹ the MXF D-10 format has the specificity of being able to carry up to 8-channel AES3 elements, and most MXF D-10 files available have these 8 AES3 channels. When broadcasters want to implement a D-10 file but

only want to handle one audio stereo channel, they are obliged to deal with six other audio channels even though they are not being used. In this case, managing the channel-valid flag in the AES3 stream as described in SMPTE ST331 can be necessary.⁸

Another point is that the AES3 sometimes hosts a Dolby-E-encoded audio stream—in accordance with SMPTE 337M non-pulse-code modulation (PCM) data in AES3.²⁰ In this case, most devices or software in a production workflow handle this as an AES3 PCM even if it effectively transports a 5.1. Dolby-E-encoded signal. In the codec layer, we often handle files with Dolby-E metadata wrapped in MPEG-2 elementary stream user data.¹⁶

Now let's look at how this multichannel information and the Dolby-E properties are managed in the wrapper layer. For the multichannel information, the generic sound descriptor set in MXF has a best-effort channel count property that describes the number of sound channels.⁵ We have encountered MXF files without this channel count metadata; sometimes, this metadata exists but is not set with the right value. As for audio encoding in the MXF layer, and more precisely in the case of Dolby-E, a Dolby-E-encoding universal label (UL) exists in SMPTE 400M (RP 224).²¹ This

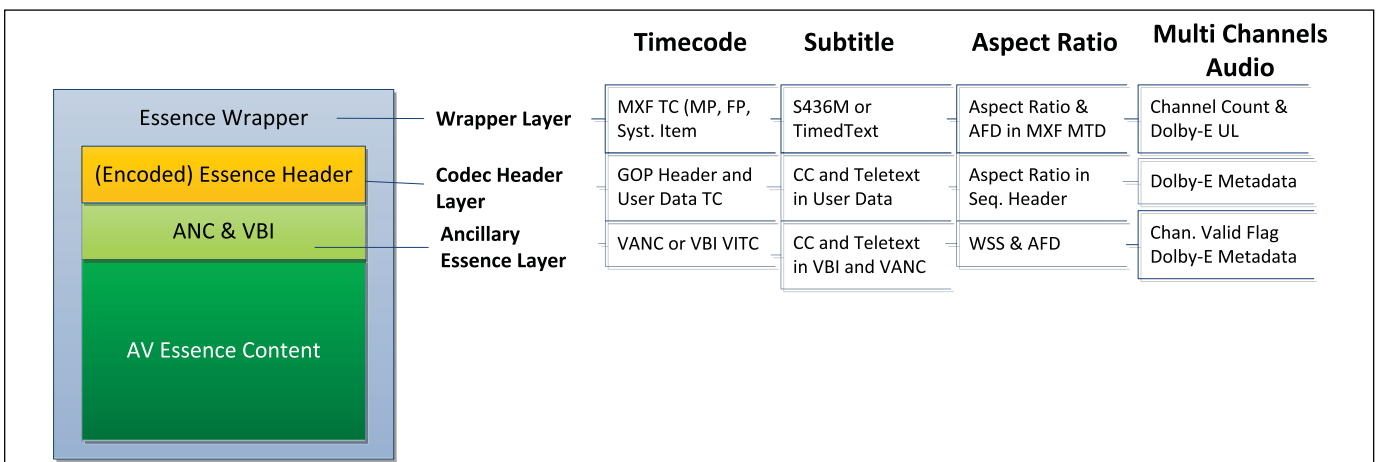


Figure 6. AV file DNA.

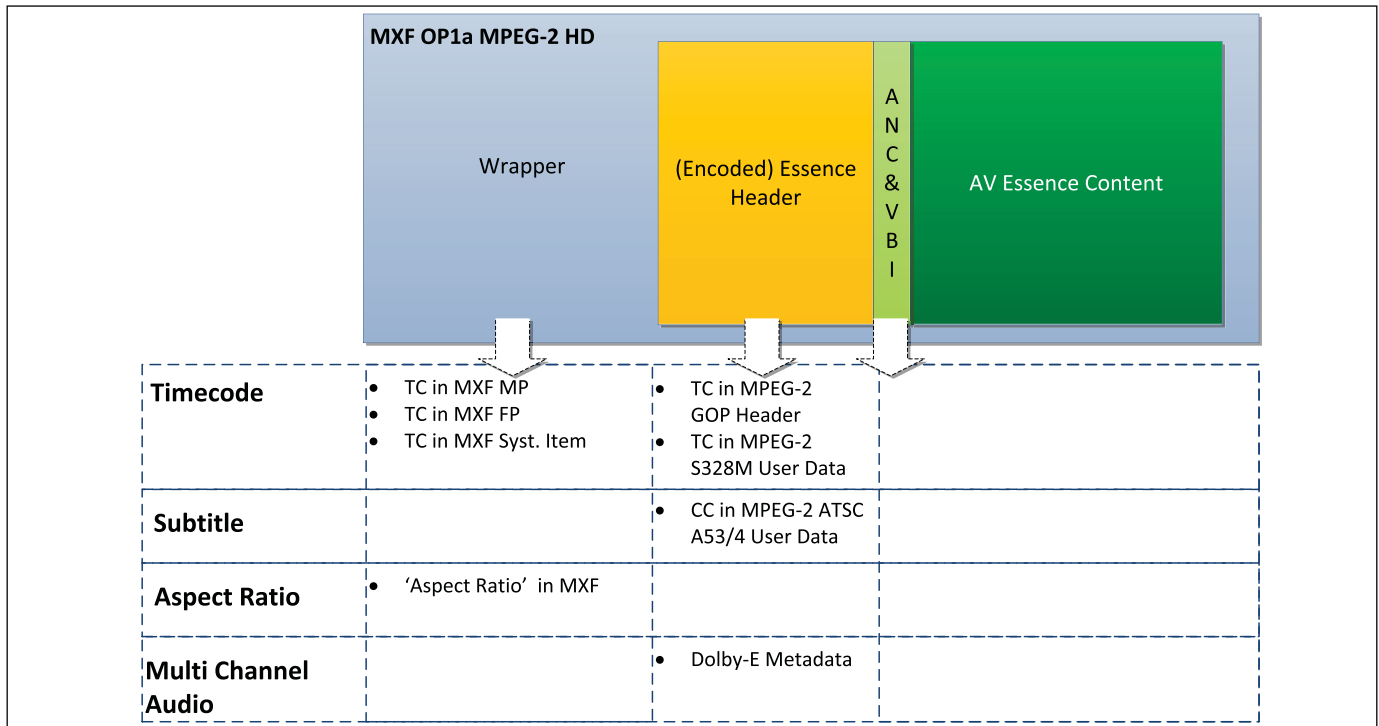


Figure 7. Implemented U.S. AV MPEG-2 MXF file.

Dolby-E UL should be used in the sound essence coding property of the MXF generic sound descriptor, but the metadata is generally never set to the correct value in the Dolby-E–embedded MXF files we have encountered.

Despite the numerous mechanisms for inserting metadata in MXF, we have never seen Dolby-E metadata in an MXF layer. This is one of the reasons it is so important to carefully manage it in the codec header, but these kinds of implementations are driven by manufacturers, which each have specific requirements for the ingest, transport, and manipulation of encoded streams and their associated metadata. For many broadcasters, this results in interoperability issues or the loss of audio metadata. For the archiving workflow, storing multichannel Dolby-E audio and betting that any kind of system will be able to exploit it 20 or 30 years from now is a risky business. Consequently, broadcasters increasingly prefer to decode the 5.1 audio streams and archive them as PCM data rather than to depend on a proprietary codec. The above is explained in **Fig. 5**.

We have briefly described two aspects of multichannel audio streams trying to coexist in a multilayer AV file. We have not considered, for example, audio channel labeling, such as describing an English audio track as English, even though this is one of broadcasters' priorities. There is still a long road ahead before we achieve comprehensive multi-audio streams with their related descriptions in an AV file.

AV File DNA

The diagram in **Fig. 6** summarizes the points we have just covered. It provides a good overview of the potential interoperability problems in modern file-based workflows, where broadcasters need to be able to rely on accurate data for cost-effective production.

Real Multilayer MXF Files

Each AV file clearly has the same shape, with the same layers. **Figures 7 and 8** represent two of the MXF MPEG-2 HD and SD files that we had to handle. It is a good practical case study of the demonstration we just made.

These two real-life examples show that the situation can vary slightly, depending on whether the format is SD or HD. In an HD MXF MPEG-2 file, there is no information in the essence ancillary layer. In addition, the codec header layer is important for storing information that cannot be stored in the MXF wrapper. In the SD MXF D-10 file, there is significant information in the inactive lines (VBI) that remains as is in the file. The aspect ratio and multichannel audio show data redundancy, either in the VBI and wrapper layer or in the codec essence layer and wrapper layer. Again, time code information is the most prevalent metadata, with multiple time codes present in multiple layers.

CONCLUSION

Even if everyone in the broadcast industry is talking about tapeless (and file-based) workflow, the source of a program remains an AV stream from a videotape or from a physical device. This AV essence has an intrinsic property embedded as ancillary data. When this essence becomes a file, the ancillary data is sometimes made persistent by the encoders, used to populate other metadata, blanked, or simply surcharged to another value.

In this paper, we show that when there are multiple instances of the same information within an AV file, a number of problems may

arise. When a property such as aspect ratio has a different value in a different location, which value should be considered as having the right integrity? A logical approach would be to rely only on the MXF metadata and ignore all other values that may exist. However, it is not as simple as that in real implementation. Other big issues that face broadcasters occur when they have to consolidate a final program made up of hybrid files. Some may have the right information; others may not.

The purpose of this paper, which is based on case studies, is to highlight some challenges broadcasters have to overcome when

implementing file-based workflows. The MXF standard is the right choice as the production file format, but despite being an amazing toolbox, it does not offer all the apparatus for storing and managing the numerous ancillary data that make up an AV file. The MXF standard is on the way to solving this issue, but in the meantime, the ancillary data are nested where they find a place to nest. We expect broadcasters to turn increasingly toward quality control file tools—not to check the quality of the encoded content but to ensure that the file has the right metadata value to be effectively managed at each step of the AV workflow.

The 10 Golden Rules

1. Determine which systems in the production chain are the “MXF generators.” These will invariably color the MXF file they generate with metadata in essence, codec or wrapper layers. Perform an in-depth analysis of the DNA of the MXF generated by these systems.
2. Always take into account the fact that an AV file holds different layers and that each layer is a potential nest for redundant metadata. An MXF file has two dimensions: the vertical dimension is key for interoperability.
3. Stick with the existing manufacturers’ MXF implementation provided that they comply with SMPTE standards, or follow AMWA recommendations to constrain the structure of MXF files (such as RDD9, or AS02).
4. If the metadata are duplicated, ensure that they all have the same value.
5. When implementing a workflow, always check that the file metadata is in order when it goes in or out of a system.
6. Make no concessions whatsoever with MXF files when it comes to standard implementation.
7. Do not try to use MXF in a “non-standard way” if your MXF SMPTE standards are not ready to convey metadata. You can always use a proprietary format or a simple XML as a temporary solution.
8. Prepare your workflow and your MAM to be scalable—the implementation of Standards in the IT world is not an exact science. Your Master File Format will most certainly have to evolve in the future.
9. Work only with companies that are proven specialists in MXF.
10. What is true today may not be true tomorrow. Perform frequent checks of your workflow. Do not upgrade a system before going through a strict rigorous non-regression phase.

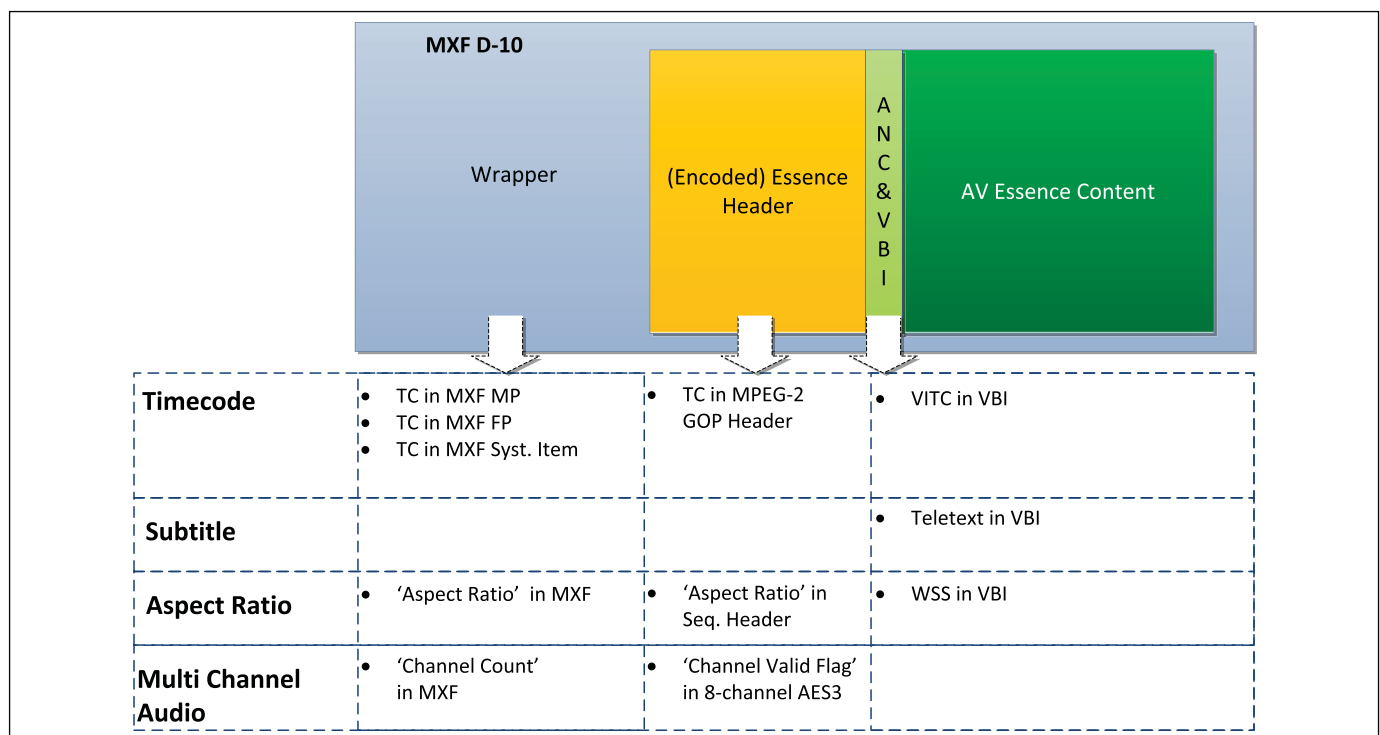


Figure 8. Implemented European AV MXF D-10 file.



REFERENCES

1. EBU Recommendation R 122, "Material Exchange Format Timecode Implementation v2.0," European Broadcasting Union, Geneva, 2010.
2. SMPTE 356M-2001, "12.65-mm Type D-10 Format for MPEG-2 Compressed Video—525/60 and 625/50," www.smpte.org.
3. ISO/IEC Standard 13818-2, 2000, "Information Technology—Generic Coding of Moving Pictures and Associated Audio Information: Video."
4. SMPTE 328M-2000, "MPEG-2 Video Elementary Stream Editing Information," www.smpte.org.
5. SMPTE ST377-1-2009, "Material Exchange Format (MXF)—File Format Specification," www.smpte.org.
6. SMPTE ST12-1-2008, "Television—Time and Control Code," www.smpte.org.
7. SMPTE 405M-2006, "Material Exchange Format (MXF)—Elements and Individual Data Items for the MXF Generic Container System Scheme 1," www.smpte.org.
8. SMPTE ST331-2011, "Element and Metadata Definitions for the SDTI-CP" www.smpte.org.
9. ATSC/53, "Digital Television Standard: Part 4—MPEG-2 Video System Characteristics," Advanced Television Systems Committee, Washington, D.C., 2009.
10. SMPTE 436M-2006, "MXF Mappings for VBI Lines and Ancillary Data Packets," www.smpte.org.
11. SMPTE 410M-2008, "Material Exchange Format—Generic Stream Partition," www.smpte.org.
12. EBU Document Tech. 3264, "Subtitling Data Exchange Format," European Broadcasting Union, Geneva, 2010.
13. W3C Recommendation, 18 Nov. 2010, "Timed Text Markup Language (TTML) 1.0."
14. SMPTE 429-5-2009, "D-Cinema Packaging—Timed Text Track File," www.smpte.org.
15. ETSI EN 300 294, "Television Systems: 625-Line Television Wide Screen Signaling (WSS)."
16. SMPTE ST2016-1-2009, "Format for Active Format Description and Bar Data," www.smpte.org.
17. SMPTE 272M-2004, "Formatting AES Audio and Auxiliary Data into Digital Video Ancillary Data Space," www.smpte.org.
18. SMPTE ST299-1-2009, "24-Bit Digital Audio Format for SMPTE 292 Bit-Serial Interface," www.smpte.org.
19. SMPTE 386M-2004, "Material Exchange Format (MXF)—Mapping Type D-10 Essence Data to the MXF Generic Container," www.smpte.org.
20. SMPTE 337M-2008, "Format for Non-PCM Audio and Data in an AES3 Serial Digital Audio Interface," www.smpte.org.
21. SMPTE 400M-2004, "SMPTE Labels Structure," www.smpte.org.

Presented at the SMPTE 2011 Annual Technical Conference and Exhibition 25-27 October 2011. Copyright © 2012 by SMPTE.

The Authors



Benoît Fevrier obtained a master's degree in computer science from the University of Bordeaux, where he acquired extensive knowledge in Digital Image Engineering. He began his professional career working on several R&D projects—in particular the 5th PCRD EU "G-FORS" project, which involved BBC, Philips, and Snell & Wilcox. In 2003, he founded OpenCube, a Toulouse-based company, which provides MXF solutions for the broadcast and D-Cinema industries. In 2010, OpenCube Technologies was acquired by EVS, a global leader in broadcast and media production systems. Fevrier is currently general manager of the OpenCube business unit in the EVS group, where he continues to carry out MXF tapeless workflow analyses, among other activities.



Valérie Popie is an engineer with a master's degree in software design and architecture from the University of Le Mans, France. She started her career at OpenCube in 2006 as a software engineer and is now product manager of the EVS-OpenCube MXF software product line, where she works, among other things, on SMPTE standardization for comprehensive MXF implementation in EVS-OpenCube Products. Popie has developed an in-depth knowledge of the theoretical and practical aspects of MXF implementation as well as significant expertise in real-life MXF-based workflow interoperability and metadata management issues.



Ludovic Dupont holds a master's degree in computer science from Toulouse's Paul Sabatier University of Science with a specialty in digital image engineering. Dupont started his career at OpenCube in 2004 as one of the very first employees of the "start-up." He is now project leader of EVS-OpenCube video server product range; his R&D activities are focused in large part on the challenges of preserving and transferring metadata from physical baseband video streams to MXF files for tapeless production workflows.