Automating Quality Control in File-Based Workflows

Primer

Table of Contents

Introduction 3
A Streamlined Workflow Approach 3

Section 1: File-Based Workflows 4
It All Begins with Ingest 4
Near-line vs. Archival Storage 6
Storage, Editing, and Mezzanine Files 7
Archiving 7
Broadcast Playout 8
Multi-Platform Publishing 9
Media Asset Management and Content Essence 10

Section 2: Anatomy of a Container 11
Meet MXF—the Material Exchange Format 11
Physical and Logical Views of the MXF Container 11
KLV Encoding 13
The Unique Material Identifier 16
Operational Patterns and Application Specifications 16

Section 3: Quality Control for File-Based Content 19
QC Methodology 19
Common Error Types 20
Automated QC: Three Alternatives 20

Section 4: Standalone File-Based Content Analyzers 21
Syntax Testing 21
Baseband Checking 21
Encoded Content Checking 22
Structural Checking 22
QC Reporting 23
MAM and Quality Control 24

Conclusion 25

Introduction

A Streamlined Workflow Approach

File-based workflows are gaining wide acceptance among networks and broadcast operators, supplanting earlier, less integrated digital workflow architectures. These file-based environments offer opportunities to streamline content production and delivery while potentially improving the quality of the end product. The file-based approach reduces operational cost and, equally important, can actually boost revenues by paving the way for new content distribution outlets. Today a subscriber can watch a favorite show on a mobile phone while sitting on a park bench.

This primer opens with an overview of each workflow component in turn and then rationalizes the whole workflow and its tools. The discussion will cover not only the functions, but also the specific tools that carry out the complex job of delivering content and services from the source to the subscriber.

The primer will drill ever downward through a software-intensive realm of file types, container formats, and Quality Control (QC) requirements. The MXF format, a leading "container" technology for internal content production and interchange, will be examined in detail, as will the role of key elements such as the Media Asset Management (MAM) system. Lastly, the need to perform automated Quality Control will be examined together with the tradeoffs associated with the various approaches and the file-based analysis tools—"standalone analyzers"—that are optimized for the task. By reading this primer you will gain a basic familiarity with the tools and technologies that support efficient and profitable modern workflows.

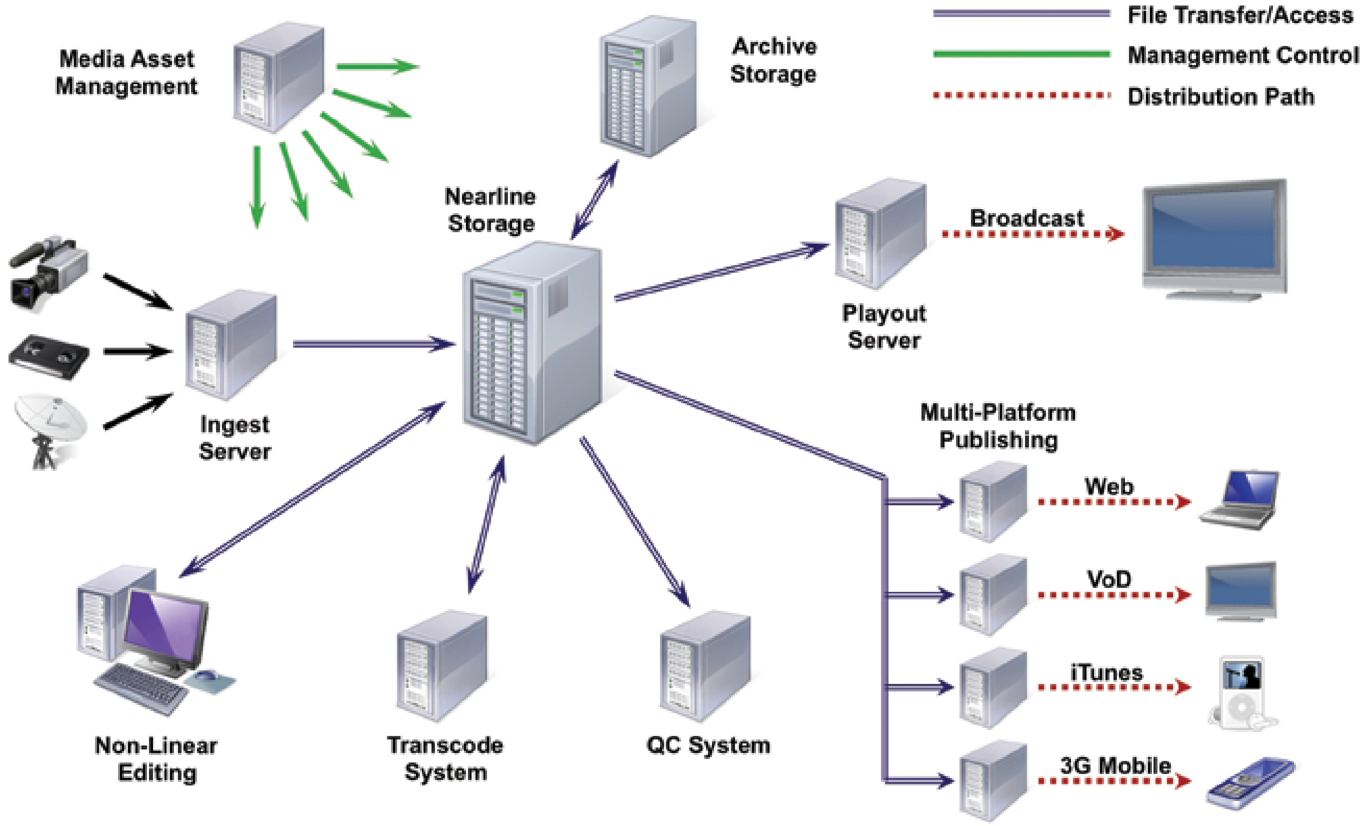


Figure 1. Architectural overview of a file-based workflow environment.

Section 1: File-Based Workflows

Every broadcast workflow is different in some way from all others, but there is a "typical" workflow configuration that forms the basis for discussion in this primer. Figure 1 depicts that flow in simplified form. After passing through an ingest server, files are transferred into and out of a central near-line storage system, which acts as a hub accessible by related functions including editing, playout, archiving, Quality Control (QC), and more. In addition there is a management system that oversees the entire workflow via networked connections (these connections have been omitted from Figure 1 for the sake of clarity).

It All Begins with Ingest

The word "ingest" is defined as "taking in" some form of consumable. Historically the term relates to food but in the workflow context it has come to mean accepting content from diverse sources, converting it to a file-based format and entering it into the larger workflow for the first time.

As shown in Figures 1 and 2, content can come from a variety of sources, ranging from a video camera to a video tape recorder to a satellite antenna. The Media Asset Management (MAM) system controls the ingest source. The ingest server exists to process the received content, where necessary, into a file-based format that includes compressed video and audio as well as time code and other metadata. The server digitizes certain types of signals and leaves others alone. For example, some video cameras record directly to a file-based format; their content can be ingested directly without conversion.

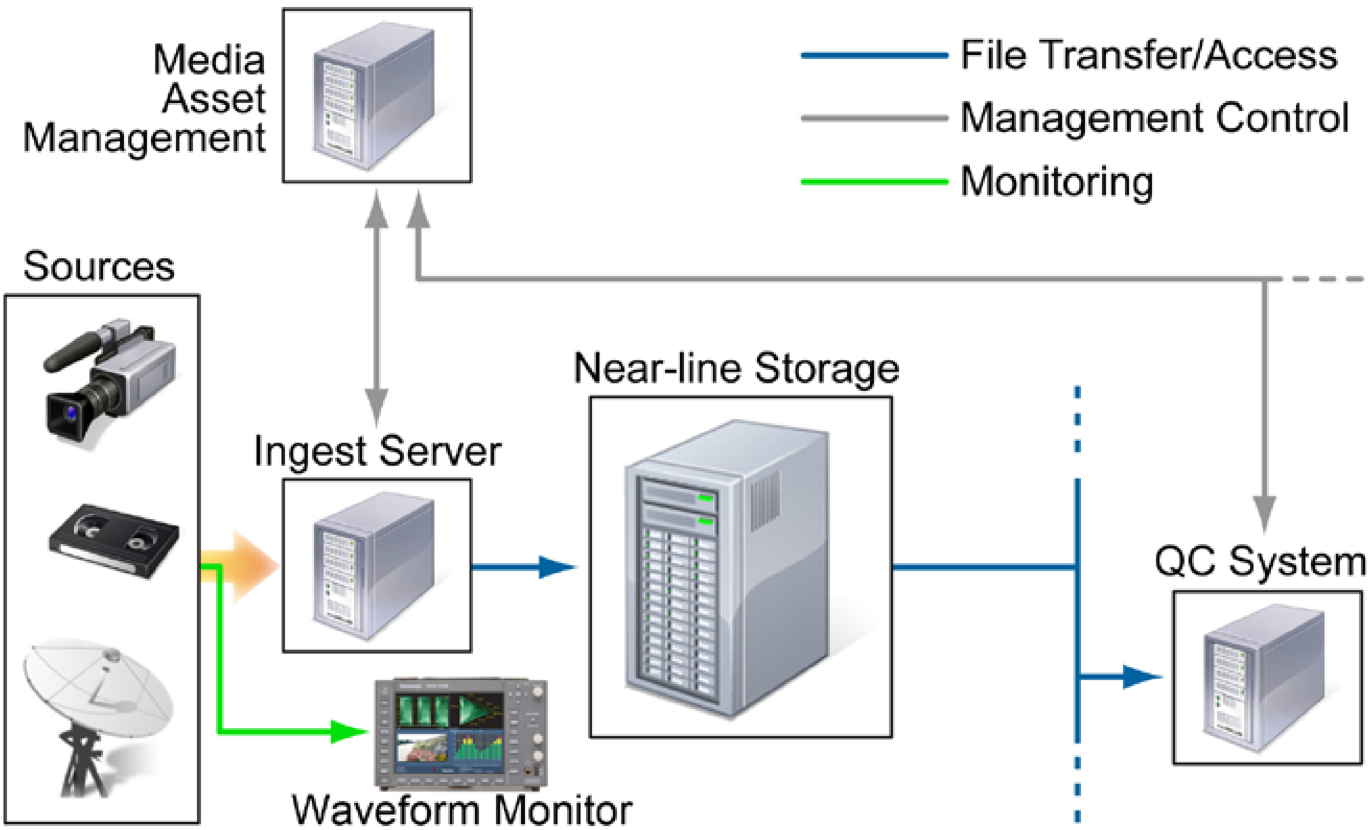


Figure 2. Ingest brings in content from diverse sources and processes it into file-based formats.

Many video cameras and other devices such as video tape recorders (VTR) source their content in the form of serial digital (SDI) signals. Incoming high-definition SDI material arrives at a data rate of 1.5 gigabits per second (Gb/s) and is typically compressed down to 100-200 Mb/s or less as it is transformed into a file-based format.

Of course, there is nothing to guarantee that incoming SDI content is flawless. SDI is a real-time interface in which content from a VTR or camera is "played back" into the ingest server. Gamut violations, color problems, and audio loudness over-ranges are among the issues that may plague these arriving SDI signals. One effective solution for these problems is to use a traditional baseband waveform monitor or rasterizer to look at the content in real-time during ingest. Ingest is the engineer's best opportunity to detect errors before they propagate into subsequent workflow processes.

Figure 2 reveals the MAM network connections pertaining to the ingest step. When content is ingested, the information about that content—program, title, play length, format, and other descriptors—must be entered (manually in most cases at this time) in a database linked to the MAM system. This allows crucial details about the content to be tracked and easily searched afterwards.

Thinking in "flow" terms, the content now emerges from the ingest server. It is possible for the first time to verify that the valid data was ingested correctly with no missing tracks and no dropouts or other picture or audio quality problems. This is where the automated quality control (QC) element swings into action. The operator can invoke QC as part of the ingest step, or the MAM can start it automatically now that the converted content is available.

Near-line vs. Archival Storage

Video files, even compressed ones, can be very large. That adds up to a massive disk space requirement for the near-line storage system and especially for archiving. But the two server types differ substantially from one another.

The term "near-line" is a contraction of the words "near online." The near-line server facilitates the high-speed movement of files between workflow functions including editing, transcoding, playout and so forth. Therefore the near-line server must support frequent, rapid file access. One common architecture for applications with these demands is Network-Attached Storage (NAS), a technology drawn from the IT world.

A network-attached storage device is simply a server whose sole purpose is to store and retrieve files. Being an IT-driven platform, the NAS uses standard network interfaces—typically gigabit Ethernet (GbE) or multiple GbE links running in parallel—in its physical layer. The protocol layer also relies on standard file-sharing protocols, such as SMB/CIFS, NFS, and FTP. These terms are household words in the IT business and now they have found a home in the video workflow as well.

The near-line server must be fast but reliability is just as important. A failure in this critical component can be very costly in terms of penalties and lost revenues. RAID (Redundant Array of Independent Disks), another technology with its origins in IT, is ideal for near-line applications. RAID combines multiple physical disk drives into a single logical disk drive. If one disk fails it can be replaced while the remaining disks continue uninterrupted to serve the original files.

A second technology known as clustering provides yet another layer of redundancy for near-line servers. Clustering involves multiple physical network interfaces with multiple physical links and multiple servers, all merged into a single logical point of access. The result is load balancing of file requests and of course, redundancy.

Near-line storage demands high availability, speed, and reliability. In contrast, archive storage systems are all about the capacity needed to store thousands of hours worth of content. Slower access is acceptable, since the urgency of searching for a historic clip for editing is usually far less than that of pulling in a segment for today's news broadcast.

The archive server may be a conventional hard drive-based system but other approaches are available as well. Given the storage volume required for archival video applications, optical jukeboxes and offsite storage can provide cost-effective capacity. The latter approach is known as the "deep archive" - a completely independent and physically separate parallel storage system that protects against data loss when a facility goes down for some reason.

Example Format	Codec	Container	Bit Rate
DVCPRO HD	DV	MXF Op-Atom	100 Mb/s
XDCAM (SD)	MPEG-2 IMX	MXF OP1a	30–50 Mb/s
HDCAM SR Lite	MPEG-4 SStP	MXF	220 Mb/s
AVC-Intra 100	H.264/MPEG-4 AVC	MXF Op-Atom	100 Mb/s
ProRes 422 (HQ)	Apple ProRes	QuickTime	220 Mb/s
DNxHD 220	VC-3	MXF or QuickTime	220 Mb/s

Table 1. Commonly-used mezzanine file formats.

Storage, Editing, and Mezzanine Files

After ingest processing, content from all sources enters the main workflow, where it is initially stored in the near-line storage system. This is the centralized storage element for the whole workflow, typically containing a cache of content intended for imminent playout—usually within the next 24 to 72 hours. Encoded files in the near-line server are also accessible by various other functions.

Looking again at Figure 1, we see that editing is one of these functions. Using nonlinear editing tools, producers or technicians piece together diverse clips to create a final output timeline for distribution.

Transcoding is also part of this “mid-stream” process. As the arrows in Figure 1 indicate, both the editing station and the transcoder acquire content from near-line storage and return it after processing. Transcoding is simply the format conversion of media assets—from an ingest format to a common mezzanine file format, or from mezzanine files to multiple distribution formats for diverse end-user platforms.

What are mezzanine files? As their name implies, they are an intermediate format - working copies that are more convenient to use than the original source material. While mezzanine files are compressed to some degree, they suffer no noticeable loss in picture quality. Being smaller files than the original source material, they require less storage space and transfer time, yet they have sufficient resolution to minimize generation loss when transcoding.

Equally important, mezzanine files are easy to edit. They are formatted with I-frames only, so that each frame is completely independent of all others. Their SDI-like 10-bit 4:2:2 sampling format provides higher resolution than that of broadcast or most other distribution formats. Any small losses in transcoding are not perceptible to the viewing audience. Table 1 summarizes some examples of commonly used mezzanine formats.

The Media Asset Management (MAM) system monitors and manages all of the activities relating to storage, transcoding, and editing. Transcoding may be invoked manually by the engineer or automatically by the MAM system. Note again that all of the active material—source files, edited files, and mezzanines—resides in the near-line storage server and is available to the system, which can re-check content quality after each editing or transcoding step.

Archiving

The archive server complements the near-line server by providing long-term storage for program material. It can store freshly-ingested content that will not be played out within the next few days, but more importantly it stores material for repeat playout in the future, and it facilitates repeat use of stored segments in new programs. Programs, clips and segments are stored with descriptive metadata, which is human-readable text describing the stored scenes.

Here again the MAM is a key component in the process. Its search functions rely on the metadata to find specific clips stored in the archive and bring them into the near-line system.

To illustrate such a procedure, imagine a news editor looking for clips that might illustrate a politician’s evolving views over the span of an entire campaign. Did today’s latest speech agree with what he said months ago? By searching on terms stored as metadata, the MAM can help answer that question. The politician’s name, the subject of the speech, the date and location of the earlier speech…all of these terms can help editors quickly access older images and content.

Clearly it is essential to create accurate metadata at the time of ingest. Like comments embedded within computer code, metadata ensures more efficient workflow operation later on. Note that archived formats may require transcoding before playout, contingent upon the needs of the content distributor.

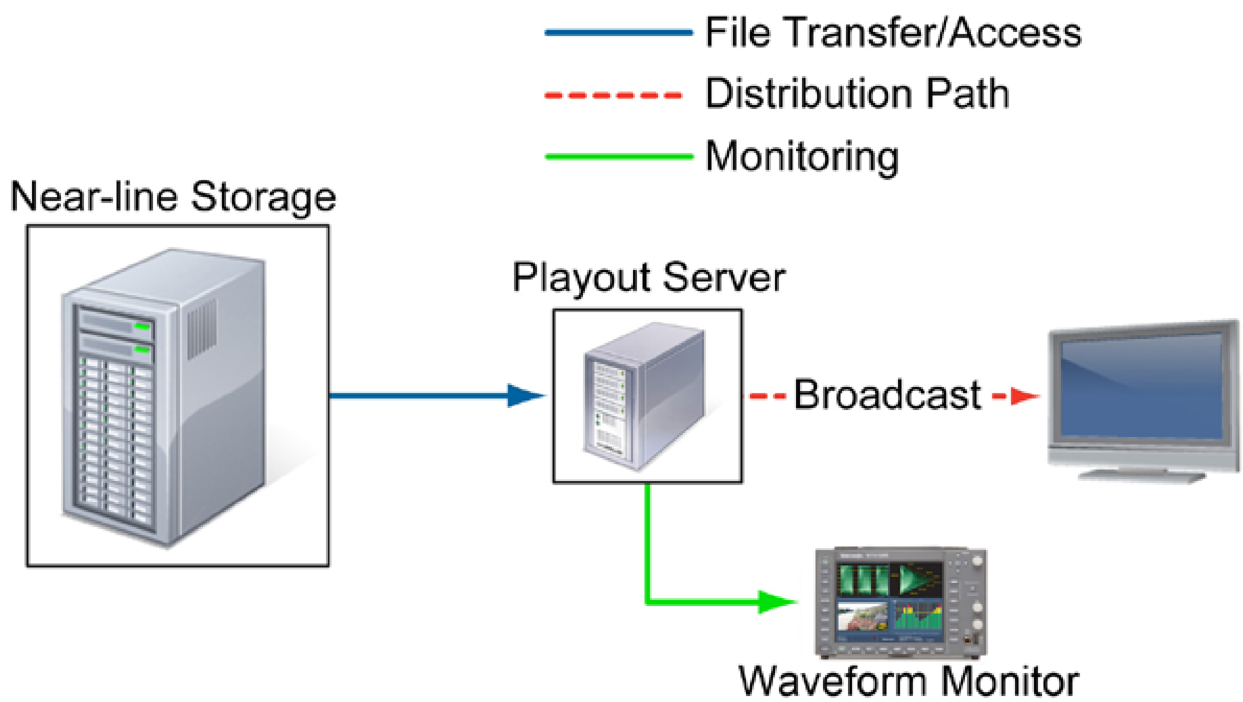


Figure 3. Broadcast playout draws on content stored for 24 to 72 hours in the near-line storage server.

Broadcast Playout

Broadcast playout brings yet another element into the workflow. As shown in Figure 3, the playout server retrieves content from the near-line system that contains the short term cache (usually 24 to 72 hours) of material.

The playout server essentially creates the distribution media. The nature of the end product depends on the format of the near-line storage as well as the needs of subsequent devices that accept and broadcast the content. If the near-line material is in the mezzanine format, the playout server requests transcoding. It may call for transcoding into an MPEG transport stream or conversion to an SDI signal.

The playout step offers an opportunity to monitor the signal quality one last time. But the monitoring tasks differ from those performed at ingest. At that point the process is concerned with content-related issues such as color out of gamut, improper picture size, and other factors that might be correctable. At playout time, qualities like the color gamut are already known but often irreversible. The object of monitoring at playout is to ensure broadcast signal compliance and confirm that the content has not been degraded in the workflow.

If the playout output is an SDI signal, then a waveform monitor or rasterizer is the right tool for examining that signal in real time and verifying the program for compliance to broadcast standards, such as the presence of closed captions. If a particular aspect such as audio loudness needs to be adjusted at this time, the waveform monitor can assist.

Similarly it is possible at playout time to observe an MPEG stream at the IP and/or RF interfaces using a dedicated transport stream monitoring system.

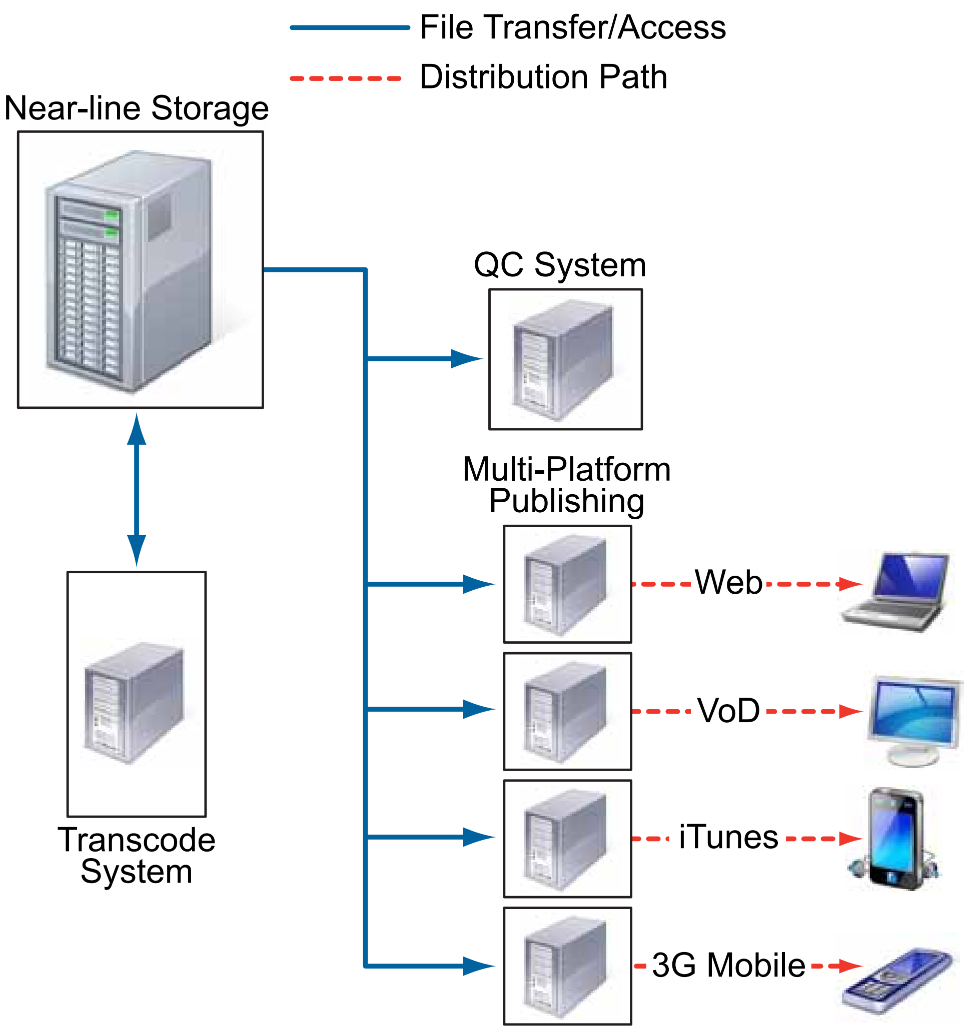


Figure 4. The transcode system converts content into formats compatible with diverse receiving platforms; the QC system checks the content quality before transmission and notifies the MAM.

Multi-Platform Publishing

Traditional television broadcasts still reach a vast audience via over-the-air transmission and cable networks. But in recent years the number of alternative content distribution paths has multiplied. Fortunately for network operators, this expansion brings with it a host of new revenue opportunities.

These alternative media include Video on Demand (VoD), streaming from web sites, transmission through 3G and 4G mobile networks, and delivery by services such as iTunes and Amazon. Creating and managing all these content delivery options is a challenge for the workflow.

With these diverse viewing environments, it comes as no surprise that the content must be transcoded into multiple and diverse codec and file formats. For example, material for 3G mobile reception must be provided at a much lower bit rate than that of the traditional broadcast channels. Here, the ability to perform quick automated quality checks can expedite this final part of the workflow. Figure 4 summarizes the interactions. The MAM, though not shown in Figure 4, manages the exchange of files going to the QC system. The QC step ensures that errors have not crept into the content due to a transcoding process that is misconfigured or operating incorrectly.

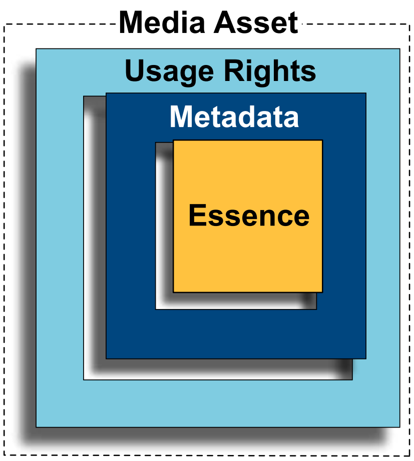


Figure 5. A hierarchy of elements makes up every media asset; "essence" is the viewer-consumable content.

Example Category	Example Data Element
Titles	Main title, Episode number, Scene number, Take number
Rights	Copyright owner, Maximum number of usages, Rights conditions
Broadcast	Broadcasting organization, Channel, Broadcast region
Languages	Primary language, Secondary language
Content Classification	Rating, Genre, Target audience, Subject, Key words
Descriptions	Annotation description, Shot description
Assessments	Award name, Content value, Cultural quality

Table 2. Examples from the SMPTE online dictionary of descriptive metadata terms.

Media Asset Management and Content Essence

The Media Asset Management (MAM) system (also known as the Digital Asset Management system, or DAM), figures prominently in the preceding workflow descriptions. We know the MAM is essentially a computer dedicated to overseeing activities throughout the process. But what is a Media Asset?

Media assets are the principal elements that are created and modified within the workflow. The heart of the media asset is known as the "essence," and encompasses video and/or audio material with supplementary metadata. By adding usage rights to the package, a complete media asset is created. Usage rights include the copyright and any limits on the playback of the asset. It is common to stipulate constraints on the playout of a particular content item to a fixed number of repetitions. Figure 5 is a symbolic view of the Media Asset elements, telescoped outward to show the hierarchy.

The need to manage media assets is pervasive across the workflow, from ingest to playout and archiving. And the key to making MAM work is the quality of the metadata stored with the assets.

There are two types of metadata:

- Structural metadata answers questions about how the file is constructed, and how to extract the individual essence components from the file. For example, some file formats interleave video and audio essence on a frame-by-frame basis, and other formats include each essence type in contiguous sections of the file. Other formats have external essence in separate files, with "pointers" to reference this content from the top-level file.
- Descriptive metadata provides details about the content. This includes both machine-generated values such as play length, frame rate, and picture size, and also human-generated keywords such as "Supreme Court" or "Paris" to facilitate future search and retrieval. Descriptive metadata also summarizes when and where the content was recorded.

An effective descriptive metadata system streamlines the search and retrieval tasks. SMPTE maintains a broad online dictionary of descriptive metadata terms (see http://www.smpte-ra.org/mdd/index.html); example elements are shown in Table 2.

Another important MAM responsibility is workflow automation. Moving assets from near-line to archival storage is a clear example of a process that lends itself to automation. After an asset has been played out, there is no need to have an operator wait for the program to finish and then move it manually from near-line storage to the archive. The MAM can handle this step automatically. The content, if properly identified by its metadata, will be easy to locate when it is next needed.

Invoking the QC function after ingest, performing QC after each transcode, and other diverse workflow functions also can be handled automatically. Workflow automation requires making products (both software and hardware) from different vendors interoperate efficiently. Workflow automation will be discussed later in this primer.

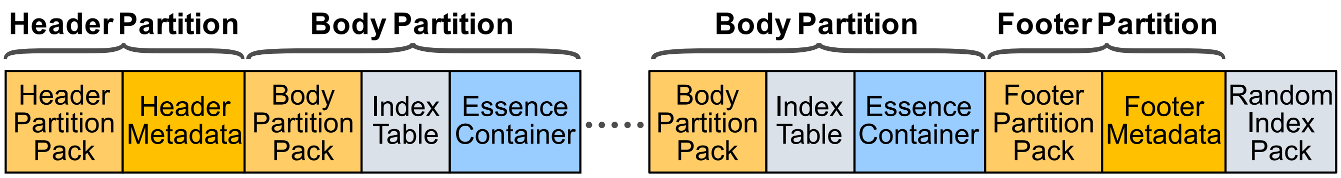


Figure 6. The MXF container is a flexible medium that delivers partitioned files that embody both essence and associated metadata, indices, etc.

Section 2: Anatomy of a Container

Meet MXF—the Material Exchange Format

By now it should be clear that content can't be allowed to circulate through the workflow as uncollated frames of data. Instead, information is packaged within "containers," also known as "wrappers", that organize video/audio essence and metadata into an expedient form. Many container structures offer flexibility in their audio/video encoding formats, supporting MPEG-2 content, H.264 content, and more. In contrast, some containers are dedicated to a particular codec type. Note also that containers may be optimized for applications such as acquisition, playback (streaming) or non-linear editing. Container formats include MPEG Program Stream, MP4, QuickTime File Format, Material Exchange Format (MXF) and more. The MXF container format was expressly developed as a standard for professional video and audio applications, and it makes a good illustrative example of the container concept.

MXF is not meant for consumer applications; it was designed to be used as an internal format for the production workflow. At its heart MXF is all about exchanging metadata. MXF containers encompass video and audio essence and associated metadata from ingest onward through the process.

Designed by SMPTE technologists, MXF has been a standard for years, with the most recent revisions occurring in 2009. Approximately 30 related SMPTE specifications have emerged to fully define details such as operational patterns (constraints on the format), essence containers, and mapping documents (to MPEG, DV, AES, etc.).

Physical and Logical Views of the MXF Container

At the highest level an MXF file is made up of partitions, as shown in Figure 6. A partition is a subdivision of the file that contains metadata and/or essence, and facilitates decoding of large files by "chunking" the content into manageable pieces. Each partition begins with a "partition pack" containing metadata that describes what follows within the partition.

The only partition that is expressly required in an MXF file is the Header Partition. This includes both structural and descriptive metadata for the content. One or more subsequent Body Partitions usually contain the essence.

Interestingly, the MXF standards provide a means to carry the essence outside the container file itself, though the process still sees it as "contained." But most commonly the essence is contained in multiple Body Partitions within one large MXF file. Inside the Body Partitions, Index Tables improve access to the content by mapping time code values to byte offsets within the partition.

The Footer Partition is optional and carries an updated version of the header metadata. The need for this information is obvious if a live camera scenario is considered. When a recording starts, a Header Partition is formed but the eventual stop time (and therefore the play length) is not known. This data remains blank, and cannot be used to inform workflow processes. The Footer Partition solves this problem by creating new metadata after the camera stops rather than going back and updating the header.

The Random Index Pack at the end of the Footer Partition is a locator (index table) for every Body Partition inside the MXF file.

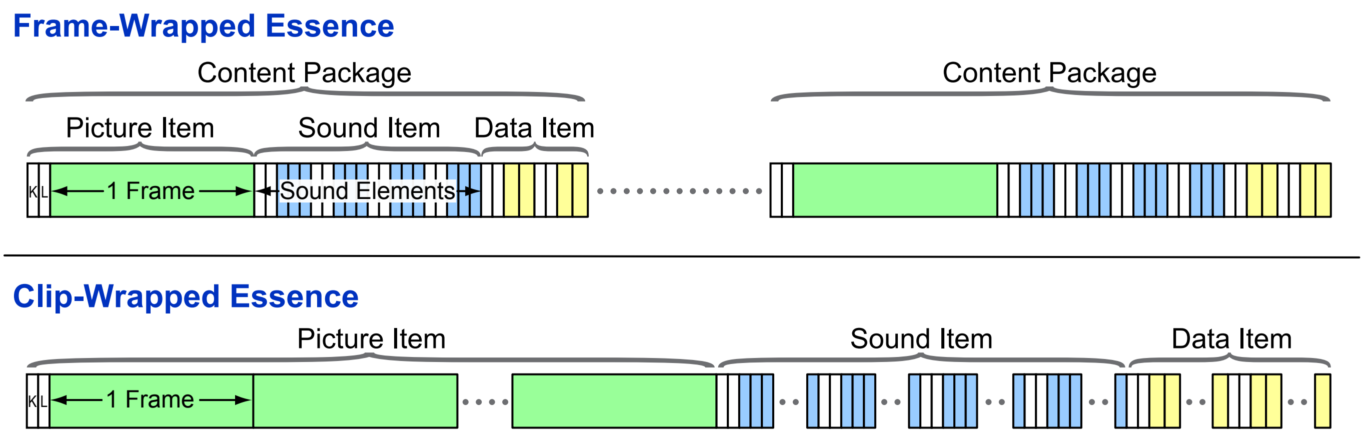


Figure 7. A view of the generic essence container format. Frame-wrapped essence is preferred for streaming applications, while clip-wrapped essence is appropriate when the content package contains just one type of essence.

The next lower level of granularity in the MXF format is the generic essence container. This entity is a sequence of Content Packages (CPs), each of which encompasses one or more of the following five items:

- The System Item which contains CP metadata and time code
- The Picture Item—the video essence
- The Sound Item (multi-channel audio typically uses multiple elements)
- The Data Item which houses closed captions, teletext, etc.
- The Compound Item which is used for indivisible essence such as DV content

Each of the Items in this list may have one or more elements, creating a structural hierarchy from content package to item to element. Figure 7 depicts this hierarchy.

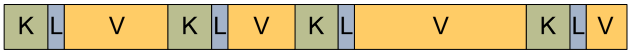
In the diagram, the Sound Item spans four Sound Elements. Each item represents one channel of sound, indicating that this stream contains a four-channel audio track such as a stereo soundtrack for two languages.

The Compound Item handles essence, such as DV, that was created with the video and audio multiplexed together. Rather than divorcing these elements into separate picture and sound items, the Compound Item is used instead of discrete Picture and Sound Items in the Content Package.

The MXF container can wrap the essence in either of two forms: frame-wrapped or clip-wrapped. In frame-wrapped essence, each content package represents one frame. The CP carries a single picture, irrespective of the compression technology; I-frame, P-frame, B-frame, etc. The frame is complete with its associated samples of sound, and captions, which makes frame-wrapped essence the preferred medium for streaming applications.

In contrast, the clip-wrapped essence carries just one content package within the container. The single Picture Item embodies all of the video, one frame after another. Similarly the Sound Elements are positioned end-to-end into a single Sound Item, as are the Data Elements, over the length of the CP. The clip-wrapped form is not optimal for playback in a streaming format, which is why it is typically used only when the CP contains just one type of essence—either audio or video.

MXF File: KLV Encoding



K = Key:

- Identifies data
- 16 bytes
- SMPTE Universal Labels
- Example: 06 0E 2B 34 01 01 01 01 01 05 02 00 00 00 00 00
  Main Title (ISO 7-bit character)

V = Value

- Essence data or metadata
- Variable-length field
- Example: 83 00 00 0C (12 bytes)

L = Length:

- Number of bytes in Value
- Variable-length field
- BER-encoded
- Example: 50 72 6F 67 72 61 6D 20 4E 61 6D 65
  "Program Name"

Figure 8. KLV triplets concatenate within the MXF file.

KLV Encoding

The next view of the MXF file is the final, close-up physical view. Here the components break down into Key-Length-Value (KLV) triplets, and ultimately an MXF file is a long series of KLV items, each of which may be composed of KLV sequences in a nested structure. Figure 8 depicts the structure.

- The K field uses SMPTE Universal Labels to announce what kind of data is coming in the V field
- The L field expresses the length (the number of bytes) in the subsequent V field
- The V field is the data value itself—usually essence or metadata

A decoder sees this file and knows exactly where to start parsing the file's contents. The key is always 16 bytes long. If the encoder can interpret the K field, it proceeds to L and then V. If not, it goes to the L field to see how many bytes it must skip in order to reach the next KLV triplet. KLV packages decode very efficiently, making MXF easily extensible. Thanks to the key field, older decoders can "automatically" skip over features they are not equipped to handle.

The KLV example in Figure 8 is intentionally simplified. Real-world KLVs can be very long, with many megabytes of data. Therefore the length field itself is of variable length, allowing it to express values into the gigabyte range.

The key field, though fixed in its length, can be optimized to make the best use of its 16 byte space. Figure 9 depicts some of these key field options. In Local Sets, one key identifies a group of values. Each of them has a one to two-byte "tag" value instead of a 16 byte key value. The tags are matched to a Universal Label (UL) by a primer pack within the header; this is essentially a lookup table that converts a one- or two-byte tag to a 16 byte UL.

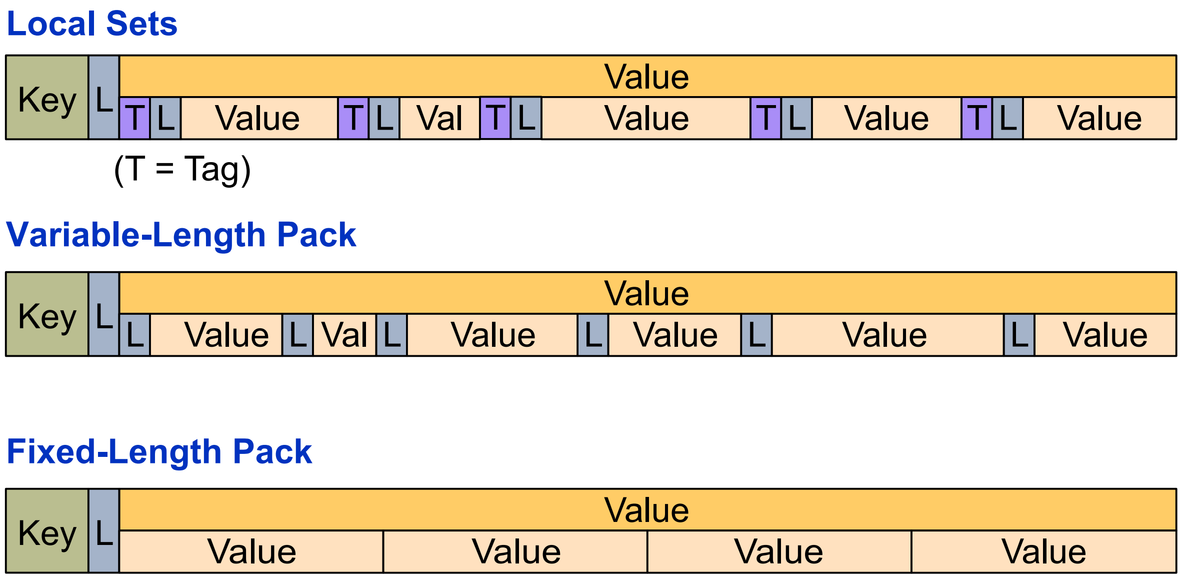


Figure 9. KLV optimization formats.

A second type of optimization is a pack made up of items that are predefined by order and position, and which can have fixed or variable lengths. The individual items do not require tags. The variable-length pack in Figure 9 contains six and only six values. One key identifies all six together and individual length fields help parse the expression and locate the position of the six values.

Lastly, the fixed-length pack in Figure 9 contains four and only four metadata values. One key field identifies all four. These values are always the same size and in the same order.

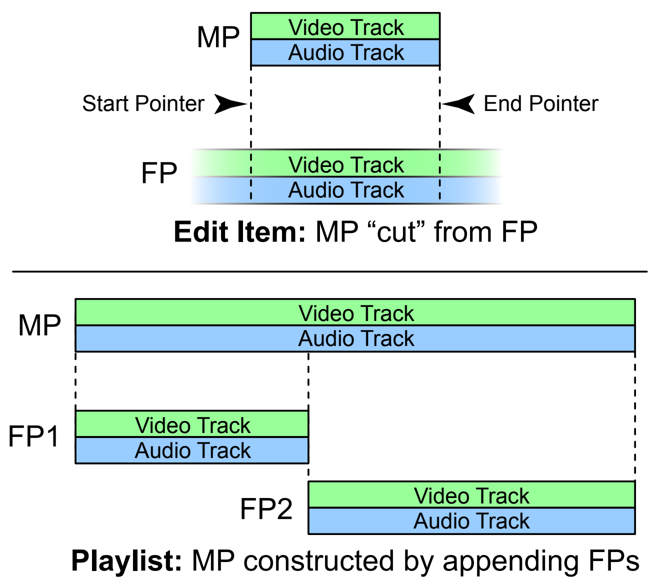


Figure 10. Material packages implement the playout sequence.

Up to this point, our discussion has focused on the physical layout of the MXF file. The physical view summarizes the order in which the various bytes of essence and metadata stream through the system.

The logical view, which describes the relationships and timelines within the essence, is equally important to understand. It is defined by structural metadata in the header partition. There are two types of logical views: the File Package (FP) and the Material Package (MP). Every package is identified by a Unique Material Identifier (UMID). Figure 10 illustrates the package concepts.

The File Package represents the input timeline of the video and audio tracks that comprise the essence. The MP embodies the playout sequence, that is, the output timeline. As Figure 10 makes clear, an MP is derived from one or more FPs.

An FP may contain several minutes of video and audio, but the MP may need just a portion of that content. Thanks to the logical construction of the MXF, it is not necessary to change the length of the whole file and "re-wrap" it. Instead, an editing tool can provide the same result simply by moving the Start and End Pointers ("Edit Item" in Figure 10).

A single MXF file may incorporate a host of concatenated file packages. A playlist defines the order in which these appear. During editing, the application pulls multiple FPs to build a single MP as shown in Figure 10.

As mentioned earlier, an MXF file doesn't have to incorporate essence; it can actually point to an external essence file. The MXF file for a playlist can be composed of metadata only, with pointers to video and audio files stored elsewhere.

The Unique Material Identifier

Both file packages and material packages are defined by a Unique Material Identifier (UMID).

Within the SMPTE 330M standard, a mechanism is defined for generating values that are locally created but globally unique. A UMID is created automatically by software or hardware, and is guaranteed to be different from any other UMID created anywhere in the world. In other words, it is globally unique.

Two types of UMIDs exist. The basic UMID is 32 bytes in length and contains the following elements:

- Universal Label (UL) expresses the material type and the methods used for creating the instance number and material number
- Instance Number (3 bytes) differentiates between various representations of material with the same material number
- Material Number (16 bytes) is an identifier developed by one of several types of algorithms. These algorithms may use timestamps, generated random numbers, and a device address such as a MAC address

The extended UMID provides an additional 32 bytes to further identify the content, for a total of 64 bytes. These break down as follows:

- Time/Date (8 bytes) based on "unit counts" such as frame rates
- Spatial coordinates (12 bytes) specifying altitude, longitude, and latitude
- Country (4 bytes) an ISO 3166-1 alpha-3 country code (e.g. "GBR")
- Org (4 bytes) an SMPTE-registered value for the organization (e.g. "BBC")
- User (4 bytes) of locally assigned data

This additional data answers the "when, where, and who" questions about the content.

Operational Patterns and Application Specifications

The final perspective on MXF architecture is the Operational Pattern (OP). An OP constrains the level of complexity in an MXF file, essentially defining how the material and file packages can be constructed. The limitations are designed to meet differing application requirements, depending on the ultimate distribution platform.

A Universal Label at the head of the partition pack announces which operational pattern is in use. This enables the MXF decoder to determine early whether it can handle the particular pattern.

Figure 11 depicts the set of operational patterns defined for MXF. Note that there are two axes of complexity, item complexity and package complexity, each with three levels. To summarize these:

- Item complexity
  - Single Item: one MP SourceClip with the same duration as the FP(s)
  - Playlist Item: multiple MP SourceClips, each the same duration as a complete FP
  - Edit Item: any MP SourceClip can come from any part of any FP
- Package complexity
  - Single Package: the MP can access a single FP
  - Ganged Packages: the MP can access tracks from multiple FPs
  - Alternate Packages: two or more MPs, representing different versions of the material (e.g. different languages, different censorship edits, etc.)

These axes equate to nine diverse combinations. A tenth possibility, OP-Atom, is not shown in the matrix but will be explained later.

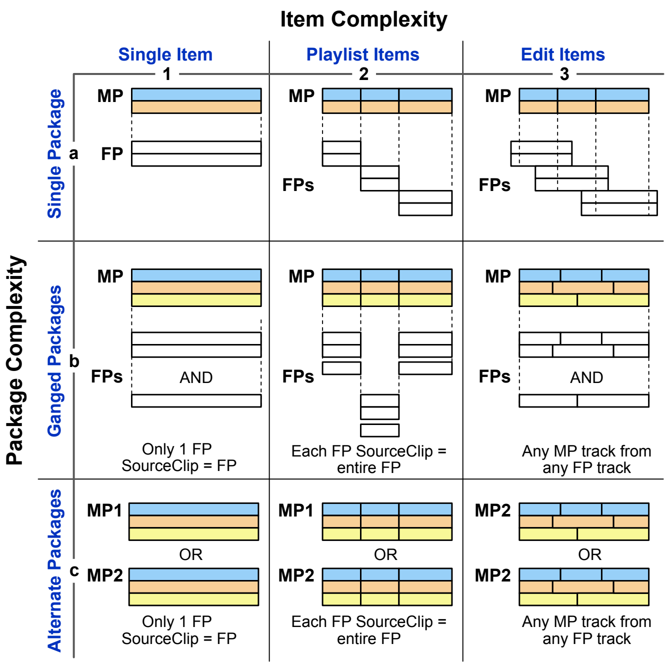


Figure 11. Operational patterns constrain the level of complexity in an MXF file.

Looking at the matrix, cell 1a (OP1a) is very simple—a single file package is the material package. The content of the FP is played from beginning to end. The patterns grow in complexity toward the lower right corner of the matrix, where the MXF file may contain any MP track composed of any FP content and may include alternate packages that offer different versions of the content. For example, an airline edit can be assembled with MXF files by using the 1c, 2c or 3c Operational Patterns, utilizing alternate packages.

Three types of Operational Patterns are most commonly used. Frame-wrapped OP1a files are self-contained and lend themselves to streaming, being made up of a single playable essence container. They embody a single item in a single package. In OP1b, there are multiple essence containers but just one material package, again simplifying the streaming of the content.

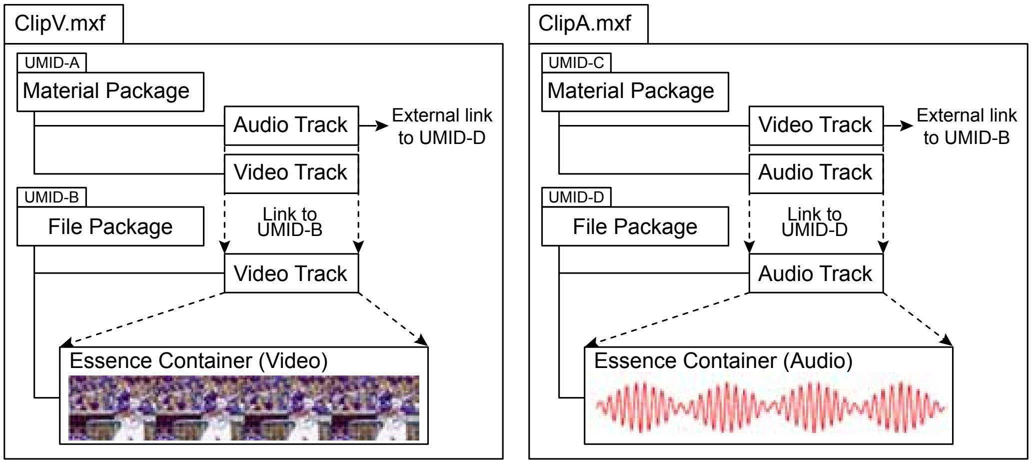


Figure 12. The structure of the OP-Atom pattern.

The third common Operational Pattern is known as OP-Atom. This specialized pattern carries a simplified representation of a single item. The file package is a single track of essence, either video or audio. Clip-wrapped OP-Atom files are often referenced from OP1b files; the OP-Atom files contain the essence and the OP1b "version file" contains only metadata. Figure 12 depicts the structure of the OP-Atom pattern.

The OP-Atom operational pattern is not constrained, so parallel OP-Atom files can share the same MP, with external links to each other. In Figure 12, two OP-Atom files have the same material package but are identified by unique UMIDs. The material package of one uses its own essence as well as the essence from the other. In effect the packages are cross-linked to each other. In most cases, however, the material package of an OP-Atom file contains the single essence track that matches the file package.

OP-Atom is most efficient for acquisition applications. A camera recording directly to MXF might capture only the video essence. The audio is recorded by a microphone working as part of a separate recording system. The resulting content is united when the individual tracks are pulled together by an editing system, and the associated OP-Atom files are referenced from the same (new) OP-1B file. Importantly, it is not necessary to unwrap the essence files during the editing process; a new version file is simply created.

An organization known as the Advanced Media Workflow Association (AMWA) has defined several Application Specifications (AS) that further constrain MXF for particular parts of the workflow in the interest of efficiency. Examples of these are:

- AS-02 (MXF Versioning) is designed for versioning and editing contents.
  - Component-based format with separate files for each piece of essence
  - Facilitates efficient mastering and editing, including multiple versions (Alternate Packages)
  - Handles any necessary re-wrapping when editing
- AS-03 (MXF Program Delivery)
  - Intended for direct playout from a video server
  - OP1a with MPEG-2 or H.264 video, PCM, AC-3 or Dolby E audio
  - Several other constraints for essence formats and MXF structure
  - Fosters a higher degree of interoperability and system reliability in multi-vendor environments
- AS-10 (MXF for Production)
  - Uses the same file for end-to-end workflows, from acquisition to delivery
  - Acquisition format = mezzanine format = playout format
  - OP1a with long Group of Pictures (GOP) MPEG

Section 3: Quality Control for File-Based Content

QC Methodology

Quality control is a crucial partner in the workflow. Historically, visual inspection has been the standard. One common QC approach is spot-checking, that is, skipping through the content by turning the tape on and checking a couple of minutes at the beginning, every few minutes in the middle, and then checking at the end. But is this visual inspection sufficient? Realistically, humans monitoring quality tend to notice just two classes of technical impairments:

- Signal level-related issues such as video luma and chroma, or audio loudness
- Obvious distortions and dropouts manifest in problems such as black sequences, frozen frames, blockiness, loss of audio, and audio/video sync discrepancies

The human eye and judgment can be reasonably effective when exposure is limited to relatively small volumes of video content. Even so, there is always the potential for error. Momentary impairments, for example, may be overlooked simply because QC operators, being human, have difficulty focusing 100% of their attention on the content 100% of the time. Moreover, there are types of errors that would be hard to distinguish even if perfect vigilance were possible.

Further complicating the challenge, the same content can be delivered in multiple ways, such as web streaming, mobile video, and over-the-top (OTT) delivery. When video is being provided over the web, Adaptive Bit Rate streams create many variants of the same file, encoded at different bit rates, which are then chopped up into much smaller files. This explosion of content repurposed onto multiple platforms means that large content networks can expect to process tens of thousands of files a month. All of these streaming files need to be tested or the content quality will be unpredictable at best. Many broadcast operations groups and content creators claim that they are simply “running out of eyeballs.”

The most practical QC solution lies in automated quality control. Automated QC processes save time and resources and are always able to devote their undivided attention to the content.

In addition, automated QC is more thorough and objective than visual inspection. It is consistent and reproducible, and able to isolate errors that are encoded deep within the files. Problems with syntax errors or encoding parameters and mismatches with structural metadata are not normally perceptible to the human eye at the point of inspection. Nevertheless they can cause serious problems later in the workflow if not corrected.

Automated quality control processes supplement human QC and can help sort out the most critical errors. This maximizes labor efficiency by enabling costly skilled QC technicians to spend their time on the relatively few problems that urgently need attention.

Common Error Types

Perceptible content errors come in many forms, with many causes. Some errors stem from problems with the original source material: incorrect color, misadjusted black or white levels from the camera, and various other baseband errors. If issues like these are not corrected before transcode, then they end up getting encoded into the file. Any attempt to correct the flaws requires decoding, modifying, and then re-encoding the material. Clearly it is best to detect errors before ingest time, so the incoming content can be cleaned up before it enters the workflow.

Problems also can occur during the initial encoding or transcoding processes if those system elements are incorrectly configured or are set to the wrong bit rate, or are set with the wrong parameters. Most compression technologies are rather complex and it is not unusual to encounter encoding errors due to these technical problems.

File-based content can accumulate errors during copying or moving from one system element to another. A common type of error is an aborted file transfer that results in a truncated file. Errors will also occur when a package of multiple files, such as an AS-02 bundle, is only partially transferred and some components are missing.

Automated QC: Three Alternatives

Three different automated QC solutions are prevalent in modern production workflows:

1. The most basic QC solutions provide only top-level information about the content's format and structure.
2. The second type offers slightly more functionality and is built into products—particularly transcoders—that are already part of the workflow infrastructure.

   When the QC function is housed within a transcoder, it is pre-integrated and well positioned to confirm that transcoding was executed correctly. But it cannot perform a comprehensive check that spans the whole system.

   Users sometimes discover that using such a QC solution within the workflow does not guarantee that the content is ready for delivery. Why? Because it is common to use the same implementation for both the transcode engine and its QC functions. A transcoding error may be allowed to pass because the QC relies on the same instructions that created the error in the first place.

   Consequently some transcode vendors are realizing that a better approach is to create a software architecture in which other dedicated QC solutions can be integrated into the processes and workflow.
3. The third type of QC tool is the dedicated file-based analyzer provided by a QC-only vendor such as Tektronix, whose Cerify® automated solution encompasses every QC point within the entire workflow. This comprehensive analyzer's capabilities and benefits extend well beyond simple checks:
   - Tektronix Cerify provides independent third party verification, and does not rely on software technology designed for a transcode engine.
   - The solution shares common heritage with other baseband video test & measurement and MPEG analysis solutions designed expressly as error-checking tools.
   - The standalone Cerify toolset reflects its instrumentation origins by delivering superior measurement accuracy and repeatability, yielding maximum measurement confidence.

The sheer volume of content that must be managed and quality-checked is driving the industry toward automated, process-spanning QC solutions. Dedicated QC platforms such as Cerify are proving indispensable in today's production environment.

Section 4: Standalone File-Based Content Analyzers

The preceding section made a case for standalone automated QC solutions. Now it is time to look at the actual test functions these tools can perform. These include:

- Syntax tests
- Baseband checks
- Encoded content checks
- Structural checks

Of course, all this activity must be digested and documented, so the final step is reporting of results. This capability, too, is part of a full-featured standalone QC solution such as Tektronix Cerify.

Syntax Testing

Syntax testing is critical as it ensures the integrity of the encoding of the file structure. Syntax issues can cause catastrophic problems, and without syntax checking it is even possible that the content won't play out at all! Syntax issues have been known to corrupt playout servers and ultimately force a channel off-air for a time. In addition, incorrect syntax can also put set-top boxes into a continual state of rebooting, or do just the opposite—lock them up and necessitate rebooting. Rigorous syntax testing can verify a compliant stream and save a truck roll.

To cite a real-world example, one VoD operator was unknowingly delivering truncated files to its subscribers periodically. No one was viewing the thousands of hours of content beginning to end to verify that it was all intact. Syntax testing for a compliant stream would have revealed the unintentionally shortened files instead of having it reported by a subscriber. In this example, Cerify was used to test thousands of files in the operator's VoD library, and its syntax checks discovered that some files lacked proper closing flags at the end of the transport stream. This error revealed the set of files that were inadvertently truncated.

Spot checks were ineffective in solving this problem, and the sheer volume of content—tens of thousands of hours—made end-to-end visual checking completely impracticable. In this (not uncommon) case, automated QC was the only solution that could solve the problem.

Baseband Checking

Tektronix Cerify can look at the decoded baseband video and audio to evaluate image and audio performance. Looking more closely at baseband errors, we see that most baseband problems are part of the original acquisition, though some may be caused by edits made in the workflow.

Gamut violations in the video are a classic example of baseband errors. For example, when content contains a "whiter than white" portion that exceeds 100% luma, it is out of gamut. An instrumented decoder can capture this violation and other gamut problems with RGB color components. Black frames, frozen frames, letter boxes and pillar boxes, and dropouts also can be detected from the decoded video content.

Audio problems such as clipping, too, are observable in the decoded baseband stream. By looking at the signal level it is possible to determine whether loudness limits, peak limits, instantaneous peaks, and true peak value limits have been exceeded, as well as long-term loudness over the span of the content. Other types of baseband audio flaws include silent (mute) passages due to audio dropouts.

Figure 13. "Slice order" errors in MPEG transport streams can cause blockiness and other picture distortions.

Encoded Content Checking

Encoders too are a potential source of errors. An encoder that is faulty can produce syntax, similar to an editing application that has flaws of its own. A misconfigured encoder or one with the bit rate set too low may over-compress the material.

One common visible effect that points toward encoder problems appears as a result of "slice order" errors in MPEG transport streams. These errors create block artifacts on the screen and degrade picture quality as well. Figure 13 illustrates the effect of an incorrect slice order.

In another case, the field order might be incorrect. Interlaced videos are encoded either bottom-field-first or top-field-first and must be played out in the same sequence. A conflict in the field order will cause distracting motion artifacts; almost a zig-zag motion when the subject should be moving smoothly across the screen. North American and European standard-definition content have opposite field ordering. It is too easy to overlook this detail when transcoding or editing.

Structural Checking

Structural checks are as much a part of QC as are baseband and encoder checks. It is possible for significant (and required) elements to be absent from content that seems syntactically correct. Or there may be a mismatch between video and audio parameters. The QC system must examine the container structure, the content, and the codec headers, comparing the actual contents with the expected information, to identify structural problems.

Checking the container structure and metadata can expose errors such as an incorrect number of streams, e.g., missing audio. This check will also reveal incorrect Packet IDs (PIDs) that are noncompliant with the CableLabs specification for VoD content for MPEG-2 Transport Streams. Content measurements can pinpoint errors such as a mismatch between video and audio play duration, or between actual and signaled bit rates

Checking video and audio codec headers may reveal "unexpected" essence formats and encoding with respect to profile and level, GOP structure, frame and sample rates, picture size and aspect ratio, interlaced or progressive scan, color depth, and color sampling. Ultimately, structural checks are all about ensuring that the content has the right format characteristics set up in the proper order.

Figure 14. This report from the Tektronix Cerify automated QC solution has captured an audio error and documented all of its parameters.

QC Reporting

Reports are a key deliverable for the entire QC process. A quality control report can be generated for each file or group thereof, and can be saved together with the content assets. The report may become part of a database or may be emailed to a QC technician who can intervene if there is a problem.

Many institutions choose to issue QC reports in both machine-parsable and human-readable formats. The former enables the MAM system to automatically extract information of interest, store it in its database and correlate it with existing metadata. This is valuable for short- or long-term analysis and record-keeping. The XML (eXtensible Markup Language) format is typically used for these reports.

Human-readable reports are often produced as PDF (Portable Document Format) files, but the same XML report described above can also include a style sheet to make it suitable for both purposes. Figure 14 illustrates such a report created by the Tektronix Cerify QC toolset. The inset box depicts an XML code fragment from the audio portion of the report. This section has several values listed, each identified by an attribute name: name track ID, units, value, etc.

The QC report provides details about errors and also presents certain measured values such as the actual loudness value observed in the file. Even if the information is not strictly classified as an error, it is still useful data that can be extracted automatically, stored as metadata in the MAM, and examined for inconsistencies with previously stored metadata. For example, if the MAM expects a program to be 44 minutes in duration but the QC system reports only 42 minutes, that discrepancy will be highlighted.

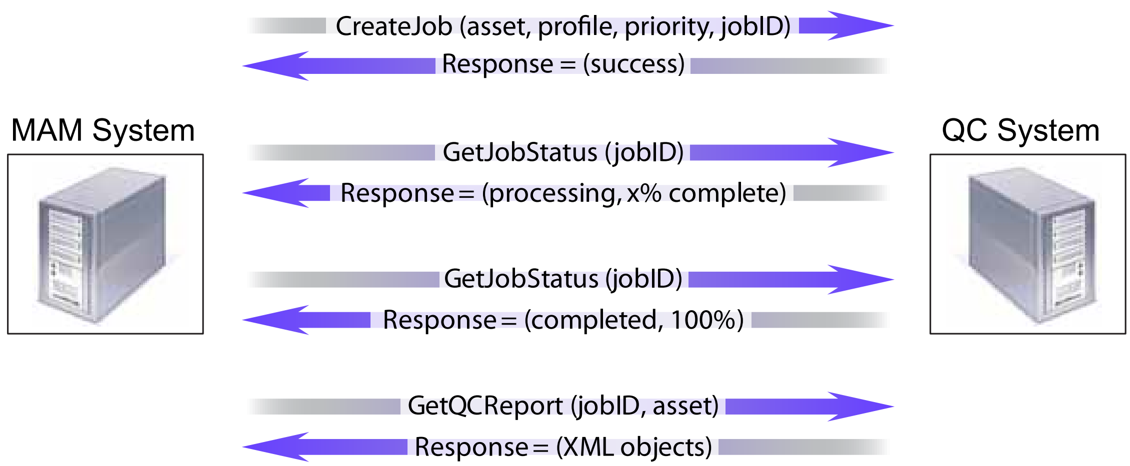


Figure 15. Web services and protocols simplify the interoperation of the MAM system and other workflow elements including automated QC.

MAM and Quality Control

It is common practice in the industry to use Media Asset Management (MAM) or automation systems to automate the flow of content through the workflow, including the QC systems. As the block diagrams in Figures 1 through 4 imply, the MAM touches most of the subsystems in the workflow: transcoding, archiving, and so forth. In effect it is an interface point between each of these software elements. This calls for an industry-standard means of integrating all these subsystems into one large system. Web services have become the preferred implementation method.

A web service simply uses existing web protocols to get the job done, for example, using HTTP to submit a request to a service provider in a server-client relationship and to receive the response. The protocol for this action is the Simple Object Access Protocol (SOAP), which resides above HTTP. SOAP is basically a way of encapsulating a web service message, both the request and the response.

Similarly, the Web Services Description Language (WSDL) is essentially a machine-parsable description of the interface. WSDL is a tool used to automatically generate library code and user documentation. It is a language-independent way of describing the program interface.

Figure 15 depicts a typical interoperation of the MAM system with a quality control system such as Tektronix Cerify, using the web service model.

In this request/response messaging scheme, the uppermost message creates a job with certain characteristics including an asset descriptor, a profile and the job's priority. This has the effect of invoking the job on the QC system. Due to the request/response nature of web services, the QC system will answer with notification of success, or failure if there is a problem with the request. The QC server never sends out unsolicited messages; It can only respond to requests.

The next step in the sequence is to check the status of the QC job. This step repeats as the server responds with data about its progress toward completion—30%, 50%, and so on. Eventually the polling is rewarded with a response of "100% complete."

As a final step, the MAM requests the actual QC report in question. The QC server inherently communicates via XML-formatted SOAP messages, so an XML-based QC report is easily sent out as part of a response message.

Conclusion

File-based workflows, when rigorously implemented, can improve labor efficiency, product quality, and the all-important "bottom line." By streamlining basic practices ranging from content acquisition and reuse to automated quality monitoring, file-based workflows can help media producers and distributors meet increasing consumer demands without commensurate increases in cost. Automated Quality Control with a standalone analyzer plays a crucial role in this. It is the only solution that can manage the explosion of repurposed content across multiple delivery platforms while guaranteeing a high Quality of Experience for viewers. With the help of automated QC, file-based processes and automation can turn today's toughest media business and technical challenges into opportunities.

Contact Tektronix:

ASEAN / Australasia (65) 6356 3900
Austria* 00800 2255 4835
Balkans, Israel, South Africa and other ISE Countries +41 52 675 3777
Belgium* 00800 2255 4835
Brazil +55 (11) 3759 7627
Canada 1 (800) 833-9200
Central East Europe and the Baltics +41 52 675 3777
Central Europe & Greece +41 52 675 3777
Denmark +45 80 88 1401
Finland +41 52 675 3777
France* 00800 2255 4835
Germany* 00800 2255 4835
Hong Kong 400-820-5835
India 000-800-650-1835
Italy* 00800 2255 4835
Japan 81 (3) 6714-3010
Luxembourg +41 52 675 3777
Mexico, Central/South America & Caribbean 52 (55) 56 04 50 90
Middle East, Asia and North Africa +41 52 675 3777
The Netherlands* 00800 2255 4835
Norway 800 16098
People's Republic of China 400-820-5835
Poland +41 52 675 3777
Portugal 80 08 12370
Republic of Korea 001-800-8255-2835
Russia & CIS +7 (495) 7484900
South Africa +27 11 206 8360
Spain* 00800 2255 4835
Sweden* 00800 2255 4835
Switzerland* 00800 2255 4835
Taiwan 886 (2) 2722-9622
United Kingdom & Ireland* 00800 2255 4835
USA 1 (800) 833-9200

*** If the European phone number above is not accessible, please call +41 52 675 3777**

Contact List Updated 10 February 2011

For Further Information
Tektronix maintains a comprehensive, constantly expanding collection of application notes, technical briefs and other resources to help engineers working on the cutting edge of technology. Please visit **www.tektronix.com**

Copyright © 2012, Tektronix. All rights reserved. Tektronix products are covered by U.S. and foreign patents, issued and pending. Information in this publication supersedes that in all previously published material. Specification and price change privileges reserved. TEKTRONIX and TEK are registered trademarks of Tektronix, Inc. All other trade names referenced are the service marks, trademarks or registered trademarks of their respective companies.

06/12 EA/ 2BW-28109-0