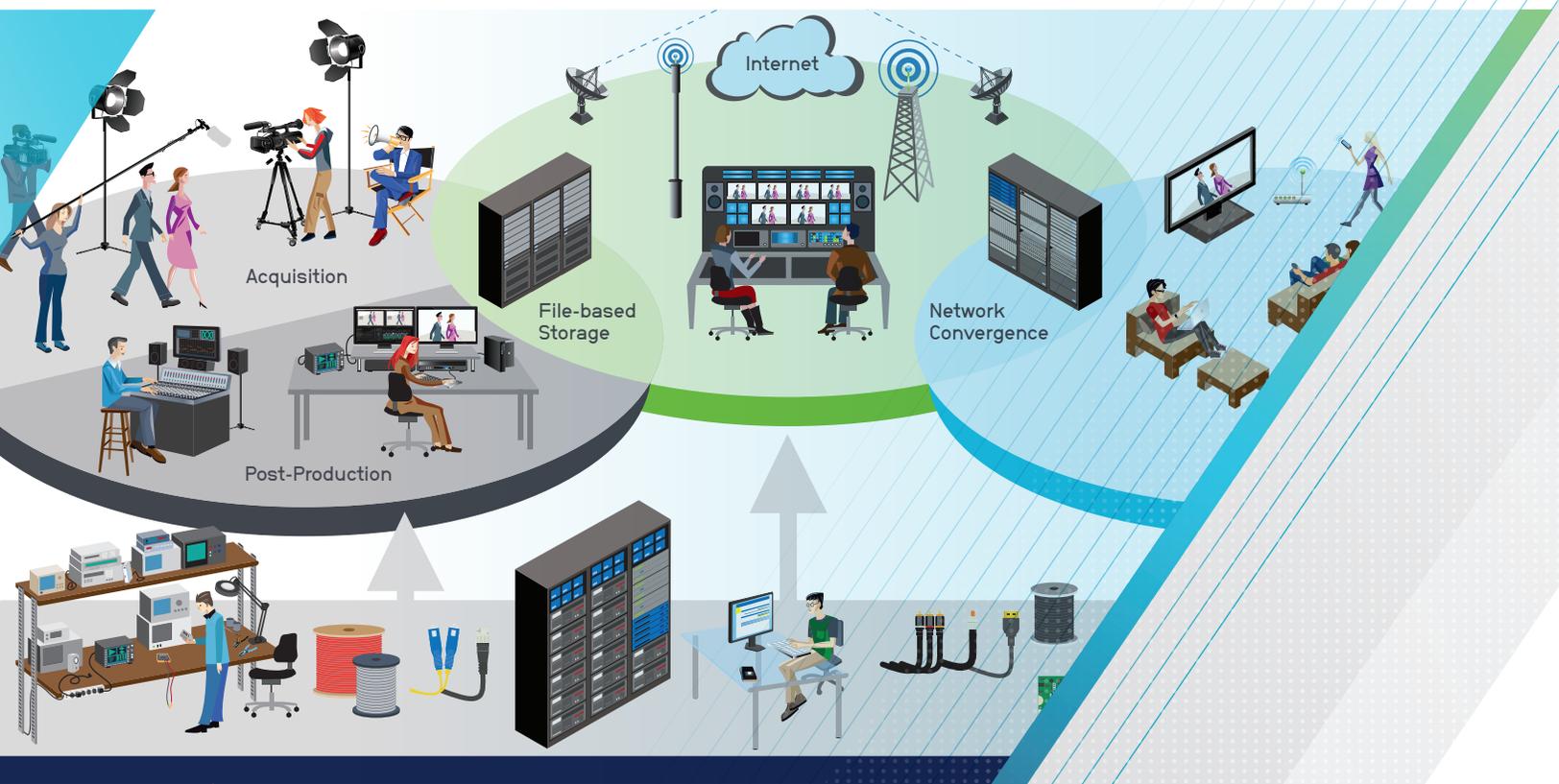


# Quality Control for File-Based Video Workflows

## PRIMER



## Contents

Section 1: File-Based Video Workflows .....	3	Section 3: Quality Control for File-Based Content .....	17
Physical Infrastructure .....	3	The Need for Quality Control .....	18
Media Asset Management .....	6	Delivery Specifications .....	18
Acquisition and Ingest .....	7	Types of File-Based Video and Audio Errors .....	19
Working with Mezzanine Files .....	8	Workflow Automation .....	20
Broadcast and On Demand Playout .....	9	Conclusion .....	21
Streaming Delivery .....	10	List of Acronyms .....	22
Section 2: Video File Formats .....	11		
Package Formats .....	11		
Material Exchange Format .....	12		
Application Specifications .....	15		
Interoperable Master Format .....	16		

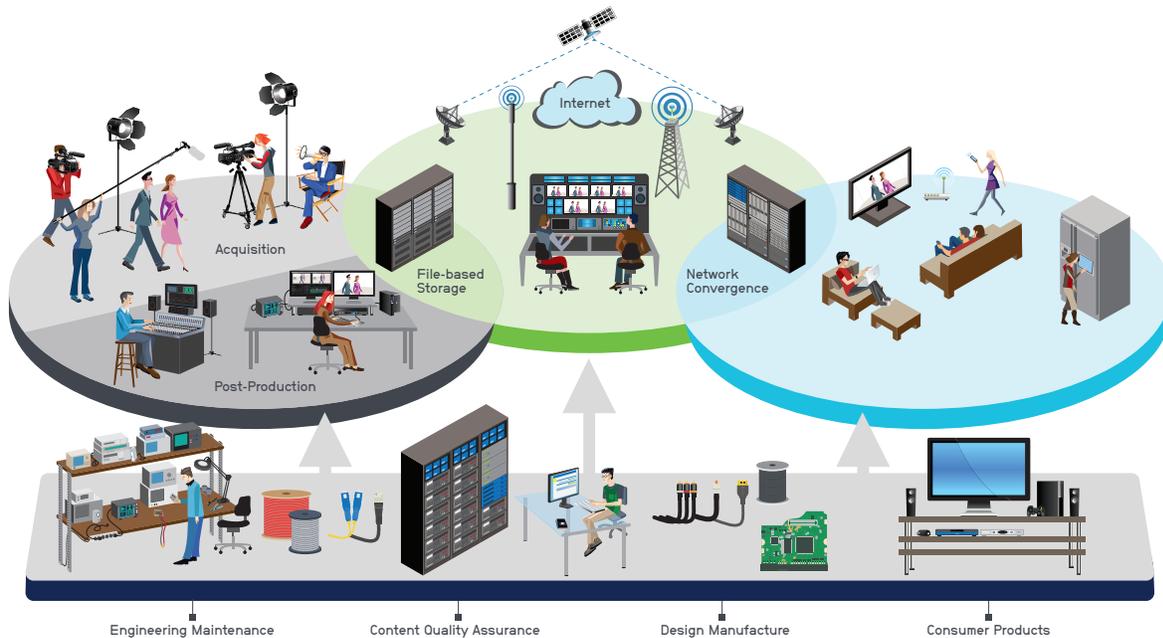


FIGURE 1. End-to-end video workflow, from the camera to the viewer.

## Section 1: File-Based Video Workflows

The Oxford dictionary defines a workflow as “the sequence of industrial, administrative, or other processes through which a piece of work passes from initiation to completion.” In the context of today’s television or cinema content, the “piece of work” is a file—or package of multiple files—that contains the video and audio essence and the related metadata for that material. A video workflow is the sequence of processes by which that content is acquired, transformed by editing and/or transcoding, and finally delivered to the end customer. A complete end-to-end workflow would include all the intermediate steps from the camera to the viewer’s display, as shown in Figure 1. In practice, multiple companies are involved, each with a workflow that implements a portion of the end-to-end solution. Well-defined specifications can facilitate the seamless delivery of content from one of these media companies to another.

A video workflow is started when the content is first acquired, either as a new capture or recording, or delivered from a content provider. At the heart of the workflow, the content may be edited to produce a different composition, and it is typically transcoded to one or more delivery formats. The workflow ends when the content is broadcast or streamed to the end viewer, or delivered to the client. Each of these functions will be explored in more detail in this section.

### PHYSICAL INFRASTRUCTURE

Most of the building blocks of a file-based video workflow are computing and storage devices similar to those found in any data center. Often, IT expertise is as important as

video expertise with respect to the design, construction, and maintenance of these facilities. Many important technologies that were developed for general purpose applications have become key components of specialized video workflows.

Storage systems used for video closely resemble those used for other data types. One key difference is that video servers tend to deal with (relatively) smaller numbers of very large files, when compared to plain file servers. Another key requirement is the need for concurrent read/write access for multiple applications working with high bit rate video files. “Nearline” (a portmanteau of “near” and “online”) storage systems are commonly used for large video servers. On a nearline server, spinning hard disk drives are idle when files are not currently being accessed, but the disks are quickly brought online automatically when file availability is required.

Video servers commonly use industry-standard interfaces and protocols for accessing files. The two main physical architectures are Network-Attached Storage (NAS) and Storage Area Networks (SAN) as shown in Figure 2. NAS devices use file-based protocols such as SMB/CIFS (Windows server shares), NFS, or FTP. NAS servers appear as externally mounted locations to the computers that need to access those video files. Network connectivity to the NAS is typically via multiple Gigabit Ethernet (or 10GbE) interfaces through a switch. In contrast, SAN devices are directly connected via a dedicated network, most commonly by Fibre Channel (FC) at speeds up to 16 Gbps or by iSCSI, which is a mapping of the SCSI (Small Computer System Interface) protocol over Ethernet interfaces. SAN devices appear similar to local disks from the perspective of the nodes connected to the SAN.

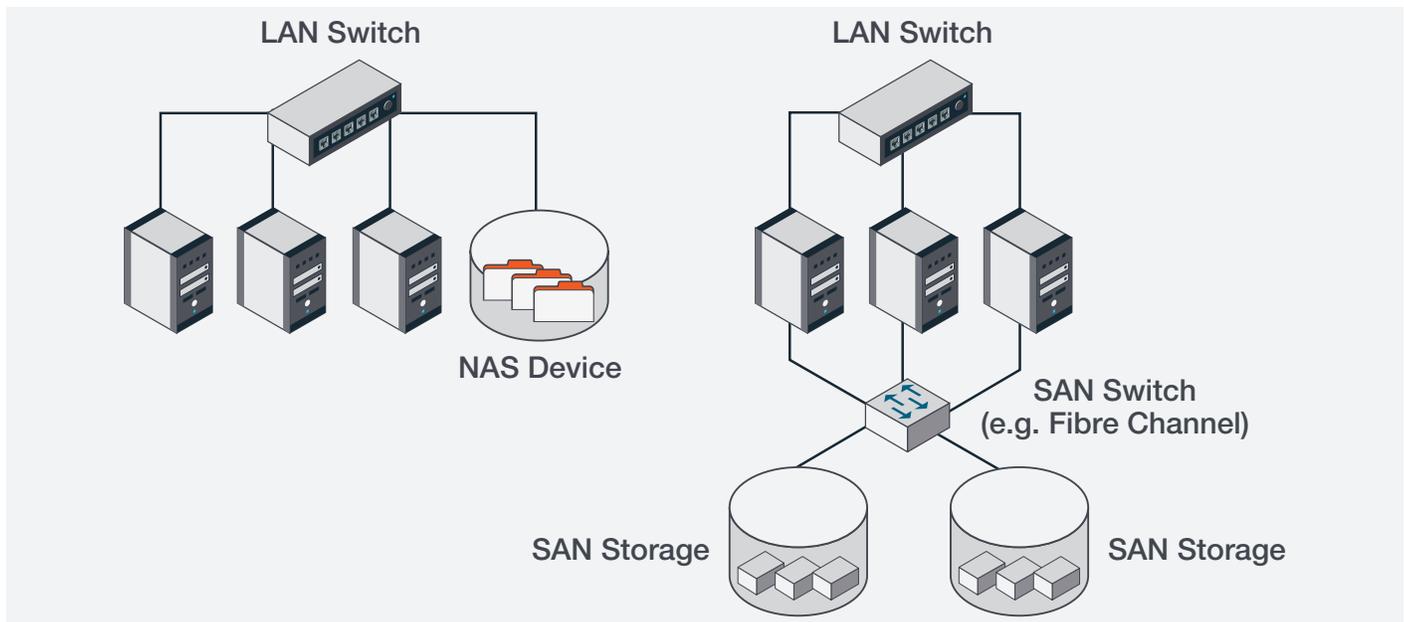


FIGURE 2. Network-Attached Storage (NAS) architecture compared to Storage Area Network (SAN) architecture.

Many video workflows—such as those in a broadcast network—are mission critical where downtime must be as close to zero as possible. Video content is usually high-value, so data loss must also be minimized. Therefore, video servers (either NAS or SAN) make extensive use of high availability and data redundancy technologies. RAID arrays of disks are widely used to guard against disk failures and can also improve performance for some configurations. RAID works by “striping” each block of data across multiple physical disks, and/or by computing and storing parity information on additional disks in the array. The parity data can be used to re-construct the original data when one or more drives fail. For example, a RAID 6 configuration utilizes two parity blocks that are distributed across all the disks in the array, alongside the data blocks. There is no stress on a dedicated parity disk because some parity blocks are stored on each disk, and the array can withstand the loss of two disks. When a disk fails, the missing data is restored on the new replacement disk by computation from the remaining data and the parity information available from the still-functional disks.

The networking architecture in a file-based video facility is critical for efficient workflow operation. Sufficient bandwidth must be provisioned to prevent any workflow operation from being interrupted or delayed. For example, a transcoder may need to read high bit rate mezzanine files (e.g. >200 Mbps) from a nearline media server, process each file twice as fast

as its play duration, and write the output files back to nearline storage. In this case, about 0.5 Gbps of bandwidth is required per file. If the workflow is expected to transcode several files concurrently, multiple Gb Ethernet or 10 GbE links will be needed. High reliability is achieved by having redundant connections—two (or more) sets of interfaces for each network node, and independent redundant LAN switches between them.

The computing platforms in a file-based video workflow are often general-purpose server-class computers. Many workflow functions—such as asset management and automation, transcoding, and quality control systems—are performed by software applications that may not need specialized hardware. Further, video workflows can often take advantage of the benefits provided by virtualization technologies, so these applications may run on virtual machines instead of directly on the same physical servers. For example, high availability can be achieved by running the application on a cluster of virtual machines (VMs) as shown in Figure 3. If any “host” server has an unplanned failure, the VMs running on that host can be re-started on a different host (one that has available resources). There is minimal effect on the application and the overall processing capacity may be maintained. A planned failover (for maintenance reasons) is even more seamless—the VM can be moved to a different host without stopping and re-starting it. The application continues to run without any interruption whatsoever.

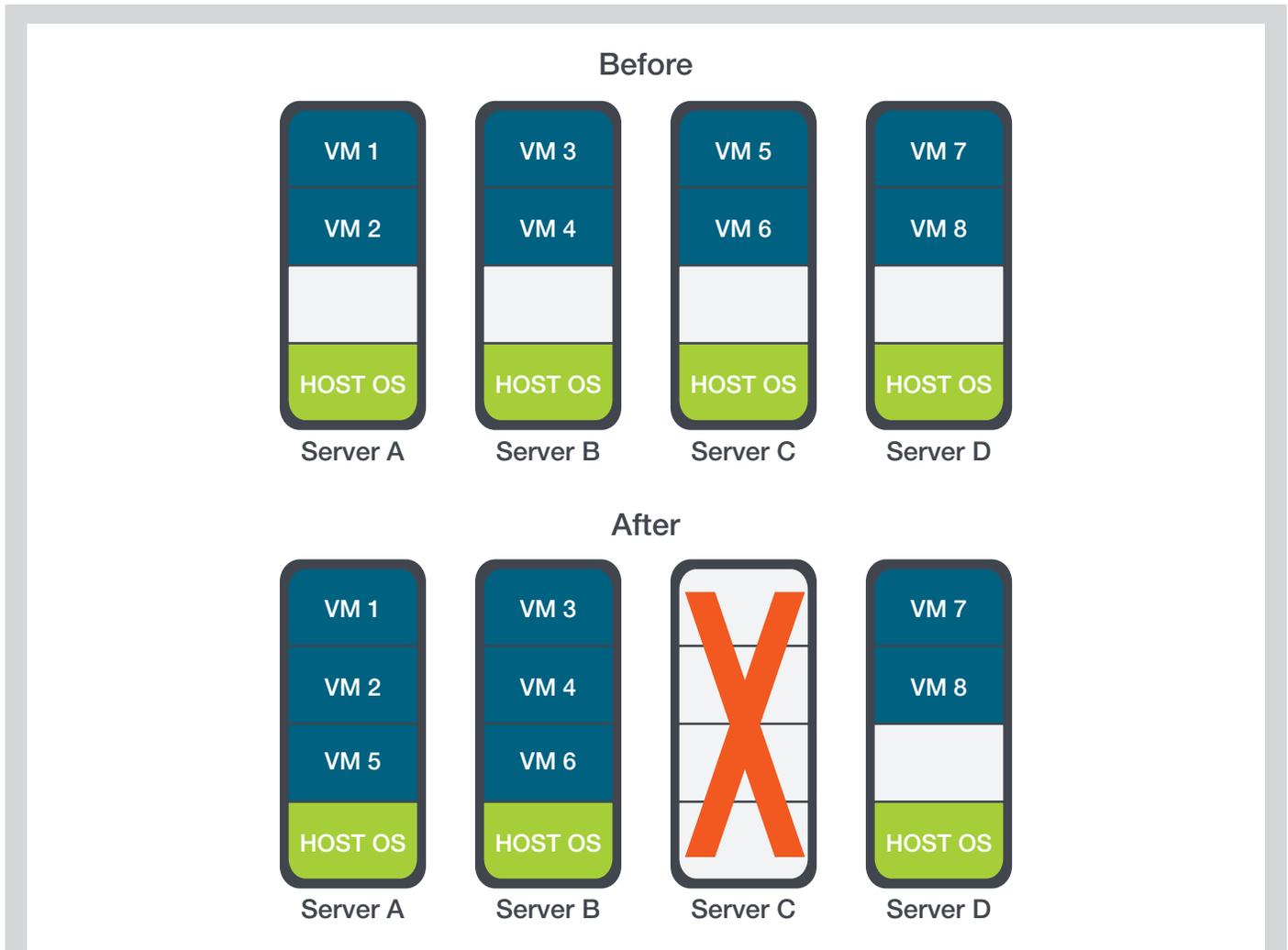


FIGURE 3.

**High Availability Cluster**

In the top diagram, the cluster is comprised of four physical servers, each capable of supporting three virtual machines. For example, these servers could have 32 CPU cores each. If the VMs are configured with 8 CPU cores, and 8 cores are reserved for the host operating system, 3 VMs could operate in the 24 available cores. With four servers, there is capacity to run 12 of these VMs, but the application configuration is set to use 8 VMs. As shown, they are evenly distributed across the four servers.

In the bottom diagram, one of the physical servers has failed and gone offline. The two virtual machines that had been running on that server are re-started on other servers in the cluster that have available capacity. The net result is that the application cluster is still fully operational because all 8 VMs are running.

If a second server was to fail, then the overall capacity of the remaining two servers would be 6 VMs, so the application might still be operational, but only at 75% capacity.

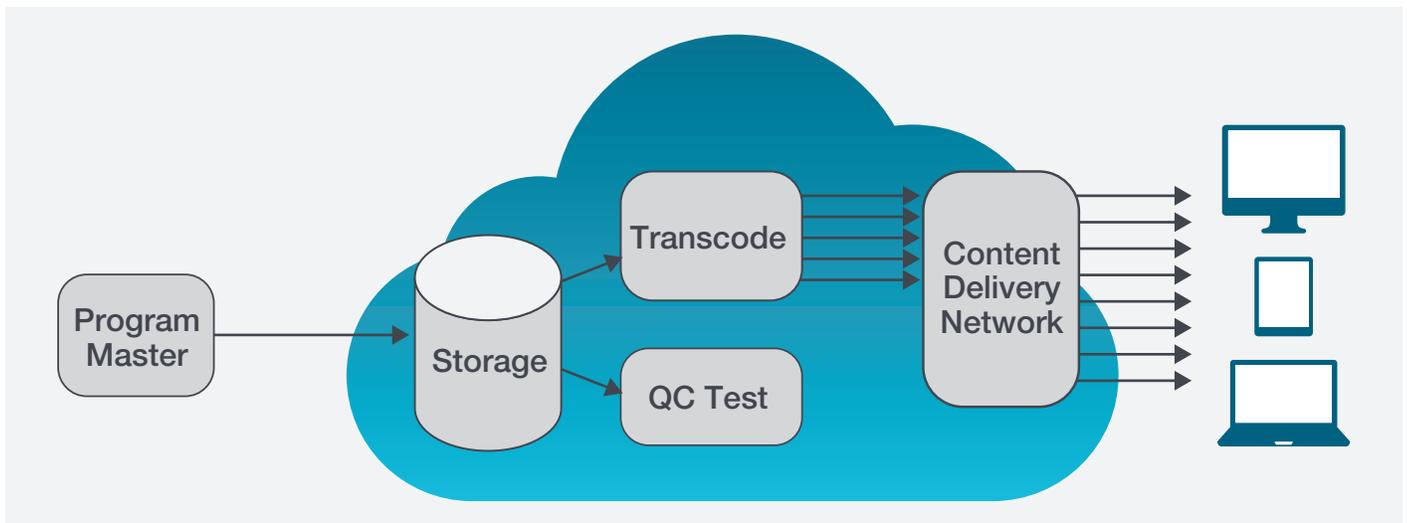


FIGURE 4. Video workflows can be partially or fully implemented using cloud-based computing and storage.

Virtual machine solutions can be implemented not only on premise with local servers, but also in the cloud with remote servers. Cloud-based solutions are attractive for video workflows for several reasons. From a business perspective, many companies prefer to spend monthly operational expenses instead of larger up-front capital expenses for server, storage, and network equipment. Ongoing management and maintenance is included in those OpEx fees instead of requiring local IT expertise. It is easier to scale up (and back down again) on demand because of changing capacity requirements, as can happen when large projects start and finish. From a technical perspective, some video workflow functions inherently use the cloud. Streaming video services utilize Content Delivery Networks (CDNs) to reach the end viewer. Further, since most Adaptive Bit Rate (ABR) architectures use multiple versions of each program (with a range of picture resolutions and bit rates), it often makes sense to perform the transcode from the master version to each resolution in the cloud rather than transcoding locally and uploading all the different resolutions to the cloud. QC testing can also be done in the cloud, pre-transcode or post-transcode as desired. Cloud-based video workflows are illustrated in Figure 4.

#### MEDIA ASSET MANAGEMENT

A critical function for medium to larger-sized workflows is media asset management (MAM). The MAM system is responsible for controlling all workflow operations from start to end in an automated manner. Operators can be notified of exception conditions, such as a file that fails QC testing or a transcode job that is queued for too long. With the potential for hundreds or thousands of files to be processed each day in some workflows, management-by-exception is necessary so that operators are not overwhelmed.

It is not uncommon for different workflow components to be supplied by different vendors. This implies that there needs to be coordination and communication between these functions, or at least between the MAM and each other function. A software interface must exist between the MAM and the file manager, transcoder, editing system, QC system, playout server, etc., as required. See Figure 5. Typically, those interfaces are developed by mutual arrangement between each software vendor, using the Application Programming Interfaces (APIs) provided by each software application, but there is also a desire to have industry-standard APIs instead of vendor-



specific ones. The Framework for Interoperable Media Services (FIMS) project is a collaborative effort to define the interfaces specific to video workflows: Capture,

Transfer, Transform, Content Repository, Quality Analysis, and Automatic Metadata Extraction.

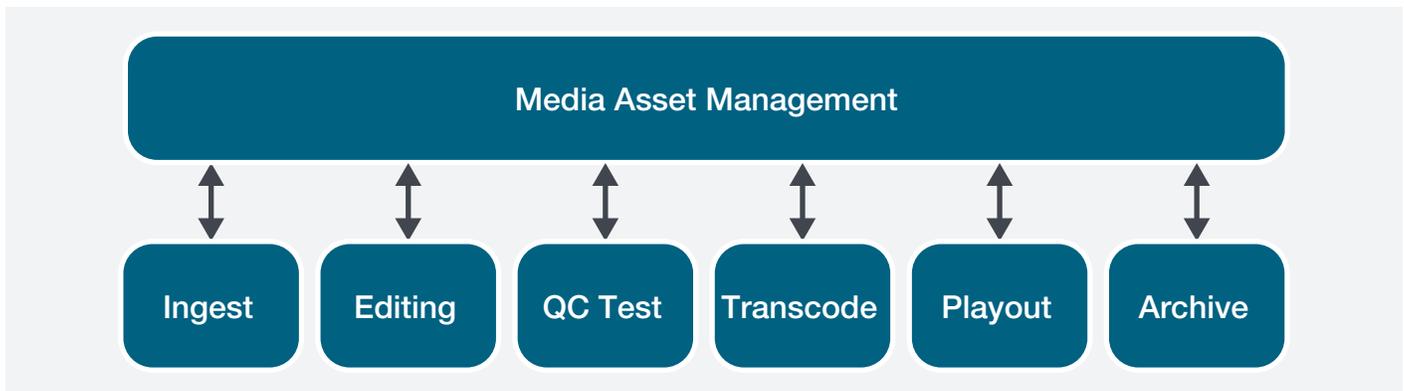


FIGURE 5. Workflow functions can be integrated with a common Media Asset Management system using a Service-Oriented Architecture (SOA).

Metadata—which is simply “data about the data”—could be considered the key ingredient of an effective video workflow. Metadata can be classified into two types: structural metadata is used to define how the individual components of the video container are encoded, and descriptive metadata is information about the content itself. Some descriptive metadata is used for technical attributes of the content (e.g. video frame rate or audio channel layout) and other descriptive metadata is used for artistic attributes (e.g. program title or release date). Metadata can be encoded within the same file as the video and/or audio essence, or it can be stored in a “sidecar” text file (typically XML format) that is part of the file package. In either case, the metadata is usually imported into the MAM’s database to facilitate workflow operations. If an editor needs to search for a particular piece of content, keywords in the descriptive metadata help find the desired files. The correct transcode profile will be selected by the MAM when the metadata for the file’s format is available. It is essential to generate the correct metadata when the file is first created and later modified.

**ACQUISITION AND INGEST**

The dictionary definition of “ingest” is to take a substance into the body by swallowing or absorbing it. The metaphor is widely used in the video production context to refer to the process of bringing content into the workflow, typically by creating a new file for that content. Content received by file delivery from a different provider could also be considered ingest with respect to the current workflow.

Files can be created directly from the camera source. Electronic news gathering (ENG) cameras often record directly to onboard flash memory, typically with some video

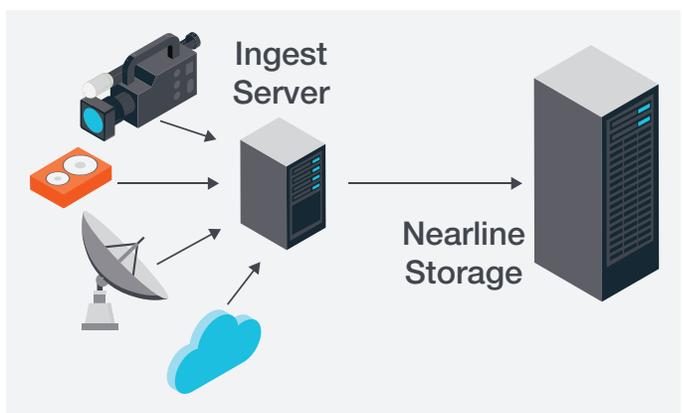


FIGURE 6. Content may be ingested into the video workflow from a variety of source types.

compression. Studio and field production cameras usually transmit an uncompressed video signal (e.g. SDI) to a separate recording unit where the file is created. For high-end productions, uncompressed (“raw”) or very lightly compressed (very high bit rate) video is recorded. Files created as part of the field or studio production will be later transferred to the nearline storage system for subsequent editing.

Previous-generation television material that was recorded on tape is often ingested into a file format, to archive the program material more efficiently or to re-purpose the content into new programs. The video tape recorder (VTR) output will be composite or component analog video for very old analog tape formats, or serial digital video (SDI) for digital tape formats. In either case, the video frames and audio waveforms are re-encoded into a new file container using the desired codecs. Many file formats share the names of the tape formats from which they were derived: DVCPRO, XDCAM, and HDCAM are three common examples.

Film-based content is ingested by a scanning process. Historically, a telecine machine was used to scan film material for subsequent television broadcast. Part of the telecine process is a frame rate conversion, from 24 fps (progressive) used in film to the 29.97 fps (interlaced) used for North American television, for example. With the rise of digital cinema production and distribution, a digital intermediate (DI) process is now used for most motion pictures. Content is usually directly acquired digitally, but scenes captured on film still need to be scanned. In the DI, each individual frame is often saved in its own image file, most commonly in DPX or TIFF formats. These uncompressed files result in an enormous amount of data. For example, a 120 minute motion picture (at 24 fps) requires 172,800 image files. For 16-bit RGB data and a 4K picture size, each file is about 26 MB in size, or about 4.5 TB for the single program.

Files can be ingested into some workflows by delivery via a file transfer service. This part of the workflow is usually automated via specially configured watch folders on both the sending and receiving side. For example, when a program has finished post-production and is ready to be delivered to the network, the file package is placed in an outgoing folder owned by the content provider (either on premise, or in the cloud). This initiates the file transfer to one or more delivery locations, where the files are received in a dedicated delivery folder. The appearance of the new file is noticed by the MAM or automation system, which then ingests the file and begins local workflow processing. Traditionally, the File Transfer Protocol (FTP) over TCP/IP was used to send large video files, but much more efficient protocols are used today.

#### WORKING WITH MEZZANINE FILES

A “mezzanine” in the field of architecture is the middle level in a theatre (between the floor and the balcony), or the middle floor at the base of a building (between the lobby and the first floor). In a video context, a mezzanine file is used in the middle of the workflow. It is often a transcoded version of the ingest format, and also different from the output format(s). For example, uncompressed video might be captured at the camera output. It is transcoded to a mezzanine format for convenience during post-production. Later, the mezzanine file will be transcoded again to the lower bit rates used for broadcast and streaming delivery.

Mezzanine files have several attributes that make them useful for post-production work. The video is lightly compressed so that the files require much less storage space and network bandwidth compared to uncompressed video files. For example, mezzanine formats for HD content typically use 100-200 Mbps for the video, which is a much lower bitrate than 1.485 Gbps of HD-SDI. Although there is about a 10x reduction in file size, the picture quality is not noticeably different to the human eye. Mezzanine formats also use complete intra-coded frames (I-frames) exclusively instead of also using predictive inter frames (B-frames and P-frames) as found in “long GOP” encoded video, making it easy to edit scenes on any frame. Lastly, mezzanine formats often use more data per pixel compared to broadcast formats, so that a superior quality version of the content is used during post-production work. Instead of 8-bit data and 4:2:0 sampling, mezzanine files can use 4:2:2 or 4:4:4 sampling with 10-bit or 12-bit data. Table 1 lists several mezzanine file formats, and Section 2 of this primer describes the respective codec and container formats.

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. Comparison of several common mezzanine file formats

Although mezzanine files are significantly smaller than uncompressed video files, they are still large files (e.g. several gigabytes of data for a few minutes of video) and can consume excessive network resources if they are unnecessarily transferred within the workflow. Nearline storage provides a central location where the file can be accessed by editing, transcode and QC systems. Lower resolution proxy files can be used in place of mezzanine files for editorial review and quality spot-checks. The MAM can automatically invoke the transcoder to create the proxy versions so that they are available.

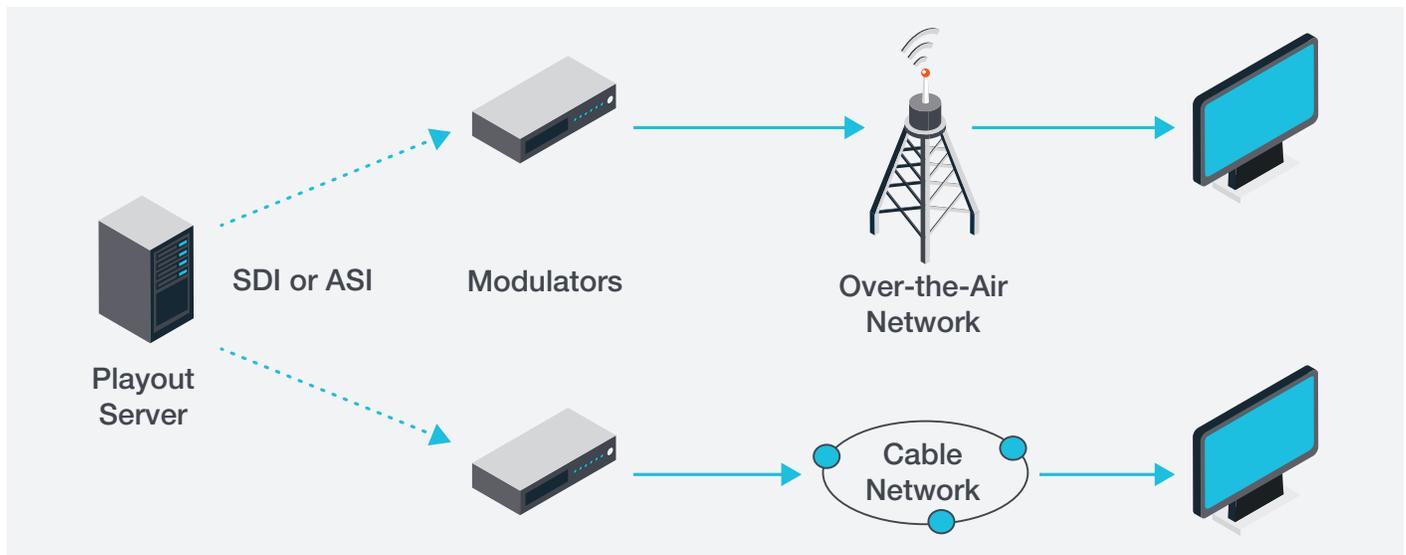


FIGURE 7. Over-the-air (OTA) and cable broadcast playback architecture.

#### BROADCAST AND ON DEMAND PLAYOUT

When the program material is ready for playout, the mezzanine master file is transcoded to a new file in the desired playout format. For example, the video is typically encoded as MPEG-2 or H.264/AVC, and the MPEG transport stream container is used. For HD content, the bit rate may be 10 Mbps or less—a further 20× compression compared to the mezzanine bit rate.

The playout server is responsible for smooth playback of these transport stream files. The MPEG TS packet data is directly output on an ASI interface so that it can be transmitted over a digital television network—terrestrial, cable, or satellite—using the local DTV standard (e.g. DVB, ATSC, or ISDB) as shown in Figure 7. Some playout servers integrate the transcode function with the output transport stream generation in a single appliance.

After broadcast, the MAM will often move the master version of the program to archive storage. Archive storage has very different performance requirements as compared to nearline storage. From a capacity perspective, a nearline system may provide 40-100 TB of storage whereas an archive system may offer several petabytes. Some archive systems utilize optical disc media that can be accessed from a “jukebox.” A mechanical arm retrieves the selected disc from its slot and mounts it onto an optical drive, whereupon the files can be copied to and from nearline storage. The access time to retrieve a file can take several seconds while the robotic arm moves, but archive operations would be very infrequent compared to nearline file operations.

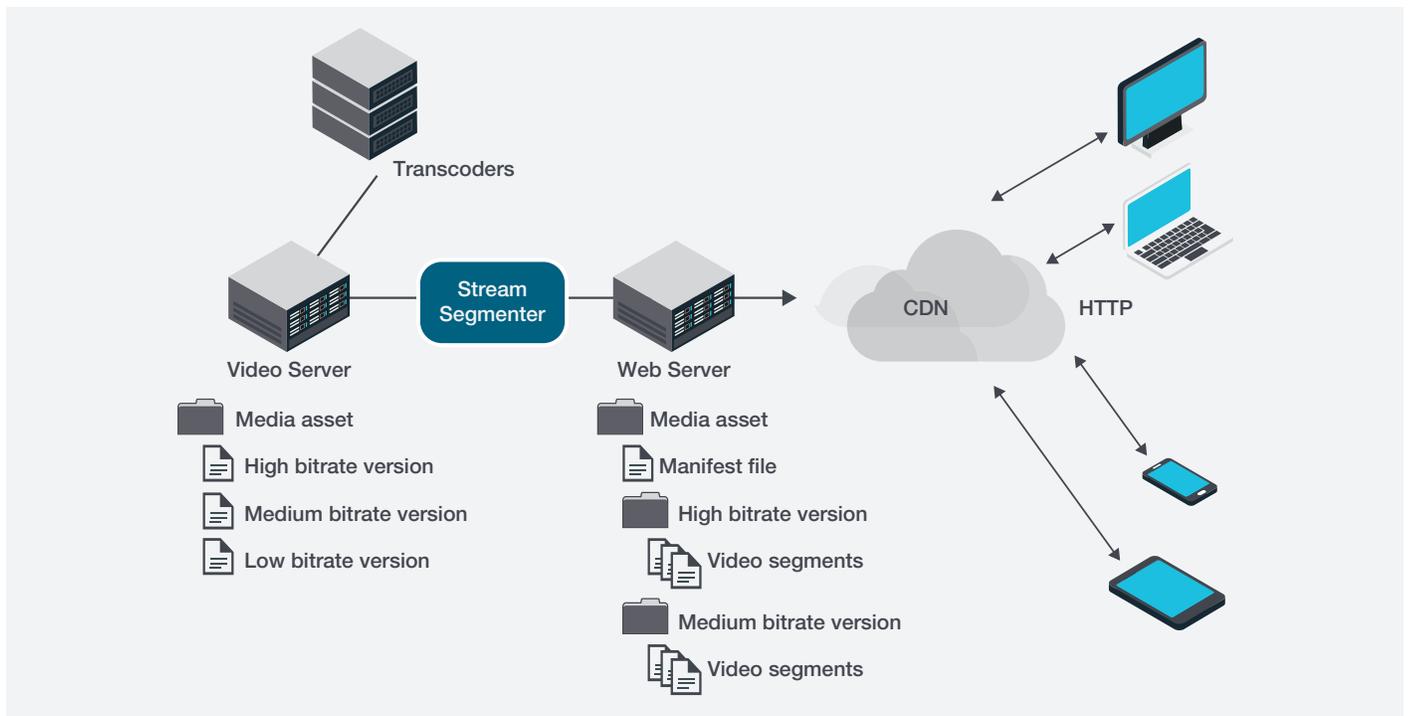


FIGURE 8. Adaptive Bit Rate (ABR) streaming video architecture.

## STREAMING DELIVERY

An increasing number of viewers now watch content via Internet connections to their mobile devices and smart televisions. Cable companies and broadcast networks are increasingly using streaming services to supplement their standard products, alongside new media companies such as Netflix, Amazon and Hulu who only offer streaming services. File-based video workflows are a natural fit for streaming delivery.

Adaptive bit rate (ABR) streaming works by adjusting the quality of the video stream in real time, based on the current bandwidth conditions of the network and the characteristics of the display device. Multiple versions of the program are encoded, each with a different bit rate and perhaps a different image size. When network conditions allow, the higher bit rate version is used. If conditions change, the stream switches seamlessly to a different rate. The viewer should experience a fast start time and little or no buffering. The ABR system architecture is illustrated in Figure 8.

A content delivery network (CDN) is used to send video streams simultaneously to many thousands—even millions—of viewers. Each program is first delivered to an origin server, where it is then replicated to many edge servers, so that the content is cached locally for better performance. The closest edge server to the end viewer (from a network perspective) will provide the HTTP stream for that program.

Delivery from the content provider to the CDN provider may be as simple as a file transfer within the same cloud service. If the master file package for the program (perhaps still as a mezzanine format) is delivered, the CDN provider will need to create the ABR package. The master is transcoded to each bit rate version, and each of those files are further fragmented into individual files of 2-10 seconds duration. This structure enables the seamless switching between bit rates—fragment boundaries occur at the same video frames in each bit rate version.

## Section 2: Video File Formats

The files used for video and audio content are often called “containers” or “wrappers.” There are many different container formats in use today, resulting from the needs of different end users and different vendors over the years. Some formats are optimized for playback and others are optimized for editing. Some formats are used primarily for professional applications, others for consumer applications, and many are used in both. The common function is that the video and/or audio essence, and the associated metadata, is encoded into a defined file structure so that working with the content is convenient and efficient.

The video and audio essence is usually encoded data, and the choice of codec is somewhat independent from the choice of container format. For example, video encoded with MPEG-2 can be wrapped in container formats such as MPEG transport stream (MPEG-TS) files, Material Exchange Format (MXF) files, QuickTime files, and many others.

Different points in the end-to-end workflow will have different requirements, which are reflected in the container and codec combinations used. Most codecs support different levels and/or profiles that are suited for different functions. For example, content may be encoded as AVC-Intra and wrapped in an MXF container during production and editing, but the same content could be encoded with “long GOP” H.264 at a much lower bit rate and wrapped in an MPEG-TS file when it is finished and available for playback from a VOD server. In both cases, the same video codec is used, but with different profiles, levels and different bit rates.

### PACKAGE FORMATS

The simple view of a video container is a single file that comprises the program content in its entirety. This is the familiar case for files recorded from a consumer-grade camera or smartphone, for example. When the file is transferred to a computer, it’s easy to open it to play the whole clip.

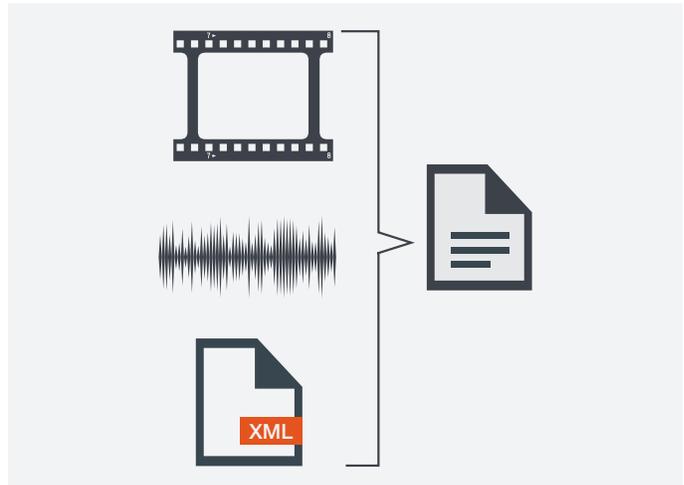


FIGURE 9. Container formats encapsulate video essence, audio essence, and data into a single package.

However, often the container for a single program does not consist of a single file, but is instead composed of a package of several (or many) files. A good example is the package typically created for adaptive bit rate streaming delivery. First, the program is encoded at multiple bit rates and display resolutions, to support a wide variety of network conditions and playback devices. Next, each of those versions is usually fragmented into individual segments of a few seconds each. The net result is that hundreds—or even thousands—of files are used for the program. Each fragment file is a standalone MPEG-TS file. For each resolution, all of the segment files are linked together by manifest or “playlist” text files that reference the segments in sequence. At the top level, a single manifest file is used to reference the individual manifest files for each resolution.

**MATERIAL EXCHANGE FORMAT**

The Material Exchange Format (MXF) is a container format defined for professional video applications. Unlike formats defined by single companies, MXF was developed by SMPTE committees representing broadcasters and equipment vendors from across the industry. The first MXF standard documents were published in 2004, but work is ongoing. Base documents have undergone several revisions and new documents are being written as new technologies are developed. To date, over 50 different specifications have been standardized by SMPTE for MXF.



One of the original design goals of MXF—and still one of its key

attributes—is strong support for metadata within the container. MXF utilizes a metadata dictionary in which standard “keys” are used to identify each metadata object encoded in the container. The SMPTE Metadata Registry (viewable online at <https://smpite-ra.org/smpite-metadata-registry>) is a list of all standard key labels for structural and descriptive metadata objects that can appear in the container. Content-related metadata such as timecode can be encoded in a similar track-like structure as the video and audio essence.

The physical view of a MXF file can be described at two levels. At the bottom level, MXF files are a sequence of KLV (key-length-value) structures as shown in Figure 10. The **key** identifies the element type per the universal label (UL) from the metadata dictionary. The **length** is the number of bytes for the **value**, and **value** is simply the data for the object itself. A single KLV could represent a simple data value (e.g. a text string for the program name), or a large complex object (e.g. an entire encoded frame of video).

At the top level, MXF files are composed of partitions as shown in Figure 11. Every MXF file must have a Header Partition, which is composed of metadata only. Any number of Body Partitions follow, containing the video and/or audio essence. Content is typically segmented into Body Partitions of a few seconds in duration (e.g. 10 to 60), making it possible to read (playback) older body partitions at the same time the current body partition is being recorded. A Footer Partition of metadata may be present after the last Body Partition. The footer metadata is often used to update values that were not known when the Header Partition was created, such as the play duration of the file.

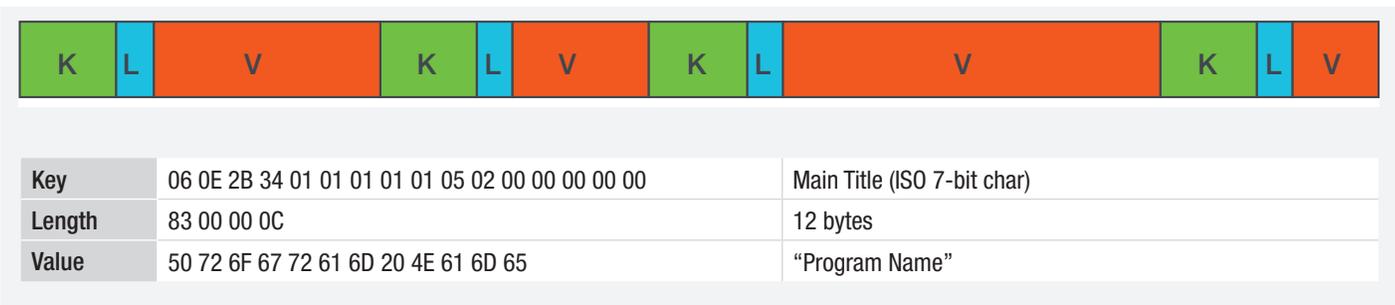


FIGURE 10. Key-Length-Value (KLV) structure used in MXF files, with an example of a metadata object.

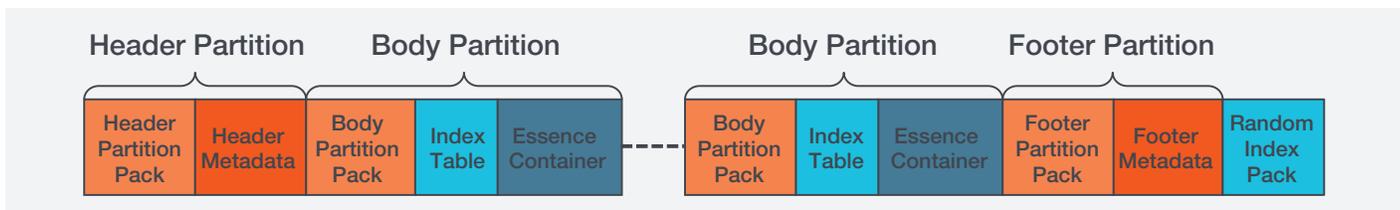


FIGURE 11. The top-level physical view of a MXF container is a sequence of partitions, including essence and/or metadata.

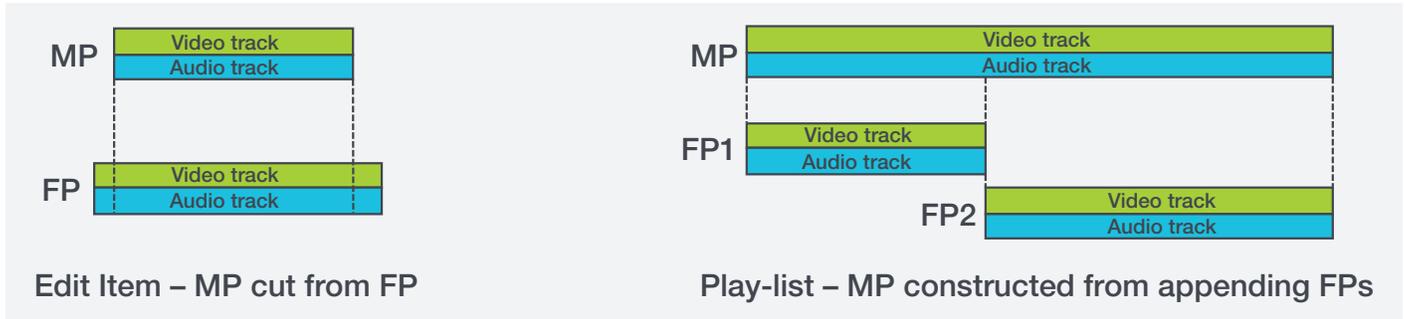


FIGURE 12. Logical view of the MXF container, illustrating how the Material Package (MP) is constructed from one or more File Packages (FP), in whole or in part.

The logical view of the MXF container is defined by two types of structural metadata as shown in Figure 12. The File Package (FP) represents the “input timeline” of the content, composed of the essence tracks present in the container. The Material Package (MP) represents the “output timeline”, or the sequence of what the viewer will see and hear. MXF allows for these two representations to differ—you do not always play back all the content in the container. An Edit Item can use an MP that is a subset of the FP. For example, a news editor might want a specific short clip to be used out of all the footage recorded by the field crew. Instead of creating a whole new file for that clip, the edit points (in and out) can be set by changing the MP of the file. A Playlist Item can be used when multiple FPs are concatenated to create the output view.

MXF defines 9 combinations of item complexity (single item, playlist items, and edit items) and package complexity (single FP and MP, multiple FPs, alternate MPs). These combinations are called Operational Patterns (OPs), and are labelled according to the item complexity (using numbers 1-3) and package complexity (using letters a-c) as shown in Figure 13. However, only two of those operational patterns are widely in use today. OP1a is the simple view of a media file, where the FP and the MP refer to the entire clip. OP1b uses multiple file packages to create the clip. The OP1b file is comprised of the Header Partition only, and references additional MXF files for the video and audio essence tracks. Those tracks files are typically OPAtom, which is a simple format that can only contain a single essence type.

		Item Complexity		
		Single Item	Play-list Items	Edit Items
		1	2	3
Package Complexity	Single Package	a		
	Ganged Packages	b		
	Alternate Packages	b		

FIGURE 13. MXF Operational Patterns (OP) from OP1a to OP3c. The item complexity is denoted by the number 1, 2 or 3 and the package complexity by the letter a, b or c.

APPLICATION SPECIFICATIONS

In the early years of MXF adoption, interoperability was an issue. Because the standards were so flexible and permitted multiple ways of encoding a file, different vendors made different implementation decisions that were incompatible with each other. The Advanced Media Workflow Association (AMWA) is an organization with representation from equipment vendors, broadcasters, and other companies in the media industry, with a mission to facilitate interoperability through standardization.



AMWA has developed several Application Specifications (AS) that more tightly constrain the use of MXF standards, resulting in a high degree of interoperability. Several application specifications have been published (see

Table 2), each representing a different media workflow, such as editing and post production, program delivery, and archive and preservation. The constraints defined in each AS include the set of permissible formats (codec types, picture size and frame rates, etc.) and required metadata.

AS-02 MXF Versioning is the application specification for a program master that supports multiple versions of the content, such as additional languages or different content edits. AS-02 uses OP1b MXF for each “version file,” which references the external video and audio essence files (each encoded as OP1a.) By separating the essence tracks into individual files, editing is a much more efficient task. For example, adding a second language version of the same program would only require the creation of new audio track files. The video track file is common to both versions and does not need to be updated. AS-02 also defines a specific folder structure for the media files and for “extra” files such as thumbnail images, QC reports and other associated files. A manifest XML file in the root

folder of the bundle contains a list of all files and folders so that applications can move or copy the entire bundle without missing any files.

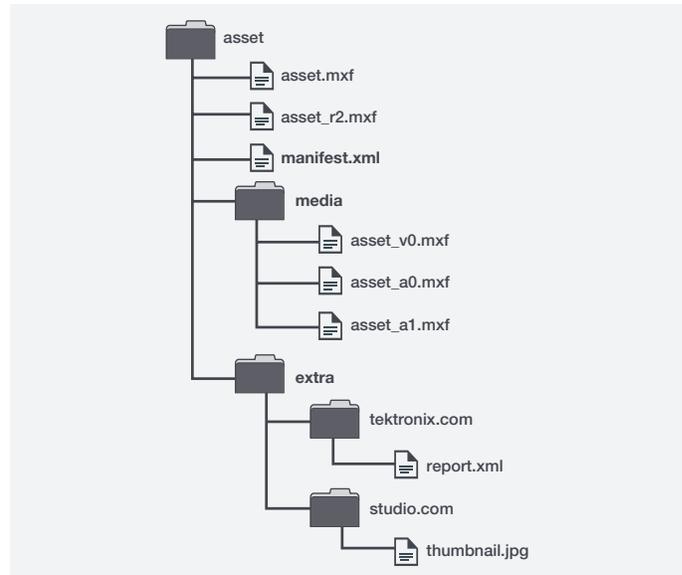


FIGURE 14. AS-02 package structure. One or more OP1b MXF version files are stored in the root location of the asset’s folder, and the referenced video and audio essence files are located in the media subfolder. Other package files may be found in the extra subfolder.

The AS-11 MXF for Contribution specification defines the formats for delivery of finished media assets from post-production companies to broadcast networks. The contribution format is not the final format used for broadcast or streaming playout, (transcoding is still necessary) but it is a well-defined and constrained format shared by both content production and content delivery companies. The Digital Production Partnership (DPP) of public broadcasters in the United Kingdom exclusively use AS-11. AS-11 uses MXF OP1a containers with either AVC-Intra Class 100 or AVC Long GOP (at 50 Mbps) for high definition material, or SMPTE D-10 (MPEG-2 at 50 Mbps) for standard definition material.

AMWA ID	Name	Description
AS-02	MXF Versioning	Packaging of program components for content with multiple versions, facilitates efficient mastering and editing
AS-03	MXF Program Delivery	Optimized for program delivery intended for direct playout
AS-07	MXF Archiving & Preservation	Used for long-term archiving
AS-10	MXF for Production	Used for end-to-end production workflow (acquisition to delivery)
AS-11	MXF Program Contribution	Delivery of finished programs at higher bit rates

TABLE 2. AMWA Application Specifications.

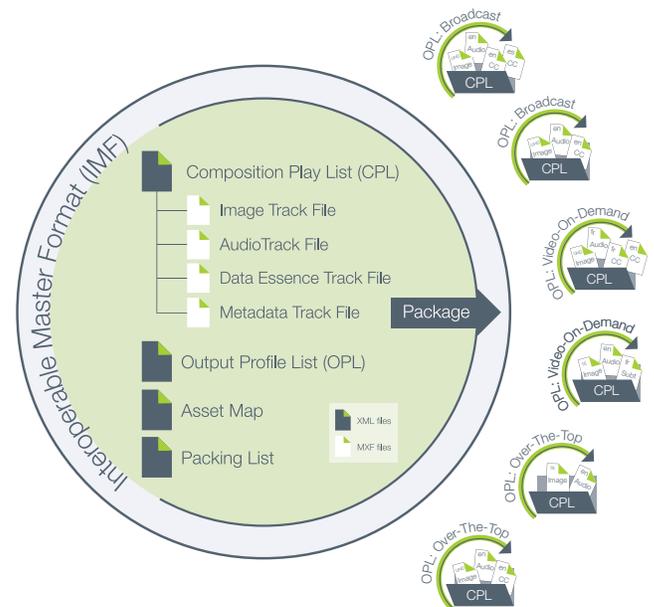
## INTEROPERABLE MASTER FORMAT

The Interoperable Master Format (IMF) was developed about the same time as AS-02. It was designed to solve a similar workflow problem—creating an efficient file package format for multiple versions of the same content—but uses a different structure than AS-02. IMF evolved from the Digital Cinema Package (DCP) format, and is now standardized by the SMPTE ST 2067 set of documents. The core framework is defined in documents such as ST 2067-2 (Core Constraints) and ST 2067-3 (Composition Playlist) and individual applications are built upon this framework, such as “Application #2 Extended” defined in ST 2067-21.

IMF addresses the problem of creating and managing many different master versions of the same material. These versions may differ in content, such as the theatrical release of a feature film, the airline edit, and the broadcast television edit. Localized versions will also have different audio and subtitles (or captions) for alternate languages and perhaps different video segments for the titles, end credits, and even localized portions of the program material itself. IMF also manages different versions based on the playout format, such as delivery by broadcast television or delivery by an OTT streaming service.

The IMF file package consists of several components, as shown in Figure 15. The Composition Play List (CPL) describes what content pieces are used to comprise one version of the program. For example, the theatrical release might include the English language titles and end credits, the entire program video, and the English audio. A broadcast television version might have some program video removed, and versions for other countries would have different audio and subtitles. Those essence files (video, audio, subtitles) use the MXF AS-02 container. Overall package size is minimized compared to using multiple complete master files because common components (such as the majority of the video essence) is included only once in the package instead of duplicated in each master file.

The Output Profile List (OPL) describes how to create different output versions by transcoding from the master essence. It includes instructions for the codec formats, picture resolution, frame rate, and more. Together, the combination of CPL and OPL define how to create a specific deliverable for a particular market. IMF uses XML files for the CPL and OPL instead of embedded metadata in the MXF file, as originally envisioned for more complex Operational Patterns beyond 1a and 1b. The IMF package also includes XML files that describe the package structure, such as the Asset Map, Packing List, and Volume Index.



**FIGURE 15.** IMF package structure. Each combination of composition (from the Composition Play List) and output profile (from the Output Profile List) determine the content and format of a deliverable file that can be generated from the IMF package.

	Application #2 Extended (SMPTE ST 2067-21 2014)	Application #3 (SMPTE ST 2067-30 2013)	Application #4 (SMPTE ST 2067-40 2016)
Picture Essence	JPEG 2000	MPEG-4 Simple Studio Profile (SStP)	JPEG 2000
Picture Maximum	3840×2160 up to 30 fps	4096×2160 - up to 30 fps with SStP Level 5 (max rate 1800 MB/s) - up to 60 fps with SStP Level 6 (max rate 3600 MB/s)	8192×6224 up to 120 fps
Sampling	4:2:2 Y'C <sub>B</sub> 'C <sub>R</sub> '	4:2:2 Y'C <sub>B</sub> 'C <sub>R</sub> ' 4:4:4 R'G'B' 4:4:4 X'Y'Z'	4:4:4 XYZ
Colorimetry	ITU-R BT.709 (8 or 10 bit) ITU-R BT.2020 (10 bit)	ITU-R BT.709 (10 or 12 bit) SMPTE DC28 DCDM (12 bit)	ISO 11664-3 (16 bit)

TABLE 3. Technical Specifications for IMF Applications

Several different IMF applications have been standardized, as listed in Table 3. Application #1 was intended to use uncompressed DPX image files in an MXF container, but this work was dropped. Application #2 uses the JPEG 2000 Broadcast Profile at frame sizes up to HD resolution. Application #2 Extended also uses JPEG 2000, but up to UHD/4K frame sizes. Application #3 uses the MPEG-4 Simple Studio Profile (SStP) at HD resolution. Application #4 is a cinema mezzanine format, supporting 8K frame sizes with JPEG 2000.

### Section 3: Quality Control for File-Based Content

Traditional video workflows primarily used manual quality control processes to check content before delivery or before broadcast. Human operators would inspect program material for visible or audible defects or other unexpected problems. Detailed QC would usually be performed in post-production, and spot checks would be performed at later stages of the workflow.



FIGURE 16. The traditional method for video quality control is visual inspection by skilled operators.

File-based workflows enable the addition of automated processes to enhance—not replace—the work of the QC operator. Files can be checked as they are created or updated in the workflow: at ingest, during and after post-production work, and after preparing the final delivery package.

## THE NEED FOR QUALITY CONTROL

Quality control is an important process at many points in the workflow. Defects that are seen or heard by the end viewer can have tangible business consequences to the broadcaster. Missed or faulty commercial spots will result in lost advertising revenue, and the “brand cost” of poor picture quality can result in a loss of subscribers. Prior to broadcast, quality or delivery compliance issues can cause program material to be rejected back to the content provider, resulting in costly rework. Typically, the earlier an issue is found, the lower the cost will be to resolve it.

Human QC operators are skilled in finding many kinds of visible and audible defects from inspection during playback. However, this approach does not scale well with respect to the large number of files and formats typically found in a modern workflow, especially when adaptive streaming video packages are considered. Operators will often only have enough time to “spot check” each file, viewing only a few minutes at the start, middle, and end of the program. Human inspection is also inherently subjective. The thresholds for rejecting bad content typically vary depending on the personal opinions of the individual operators.

Assisted QC—considering both automated and manual software solutions—mitigates many of these problems. QC software can decode and check each frame of video faster than normal playback speed, and automated QC can run continuously 24 hours a day, allowing larger volumes of content to be checked. Consistent objective results are produced from a software solution. Many types of non-visible errors, such as metadata errors, are easily detected by QC software. The end result is that the QC operator can spend most of his/her time fixing problems rather than finding them in the first place.

## DELIVERY SPECIFICATIONS

A delivery specification is a set of requirements for the transfer of media content from a production company to the receiving company (such as a broadcast network or streaming media provider.) It includes a strict list of acceptable formats and technical attributes (e.g. frame rate, picture size) for the content, and perhaps a description of minimum acceptable quality criteria. Delivery specifications are usually defined by the receiver, and it is the responsibility of the provider to meet all acceptance criteria. It is important for the provider to check for compliance before delivery to avoid rejected material.

The Netflix Full Specifications and Operators Manual (available at <https://backlothelp.netflix.com>) is one example of a delivery specification. It includes separate specifications for SD, HD and UHD (4K) formats. For HD, three options are acceptable: MPEG-2 video (80 Mbps, I-frame only) in an MPEG-TS container, ProRes 422 HQ video in a QuickTime (MOV) container, or JPEG2000 video in an IMF Application 2 Extended container. The ProRes option is useful for many content providers because it conforms exactly to Apple’s iTunes package format, so the same package can be used with both companies.

In the United Kingdom, the Digital Production Partnership (DPP) is an organization whose membership includes media companies representing the entire workflow from production to broadcasters. The DPP has published a common delivery specification that is mandatory for all content delivered since October 1, 2014. The DPP’s Programme Delivery Standard is based on AMWA’s AS-11 but also extends the specification to include mandatory technical and editorial metadata objects that must be present in the MXF file. DPP also requires compliance to several quality standards, such as EBU R128 for audio loudness and Ofcom 2009 guidelines for photosensitive epilepsy (PSE).



The success of the DPP has led to the development of common delivery standards for other countries. Variations of the original AS-11 specification are intended to be used in

Australia and New Zealand (AS-11 X2), the Nordic countries (AS-11 X3 and AS-11 X4), and in the United States and Canada (AS-11 X8 and AS-11 X9). The North American Broadcasters Association (NABA) has partnered with the DPP to develop the X8 and X9 versions. The broadcasters in Germany have adopted the ARD\_ZDF\_HFD encoding profiles (XDCAM HD or AVC Intra video in an MXF container). In France, the “Prêt à Diffuser” (PAD) specification (based on AS 10) is in use by that nation’s broadcasters and post-production companies.

## TYPES OF FILE-BASED VIDEO AND AUDIO ERRORS

Several different types of errors can exist in a video file. Many error types are visible or audible to the viewer, but a bad metadata value, for example, would only be detected by a software tool that decoded the value. Some QC tests have a clear pass or fail definition: is the container type acceptable, or is the frame rate correct? But other QC tests can be very subjective; the amount of visible compression artifacts that are present before picture quality is deemed to be unacceptable can be an opinion that varies by viewer.

The simplest types of errors to detect by QC software are related to the attributes of the file, such as formats and metadata values. Some checks can be determined immediately, such as the video codec and its profile and level. If a network's specification mandates H.264 video with High Profile @ Level 4.1 and the received file is a different format, it can be rejected immediately. Other QC checks may require measurements to be made. If the play duration of an ad spot must be 30 seconds (perhaps with a tolerance of  $\pm 0.1$  seconds), the number of frames in the clip must be counted. The QC system can also verify that measured values and attributes match the corresponding metadata value encoded with the file. Playout issues might arise if the video frame rate is actually 23.976 fps but the header metadata claims it is 29.97 fps.

Errors in how the file is encoded will be detected by a complete decode. At the container level, structural errors will often prevent the file from playing properly. Syntax errors in the video or audio tracks will usually result in visible or audible defects, although set-top boxes or player applications will try to conceal these errors. The decoder in a QC application will report syntax errors instead of attempting to hide them. Correctly encoded video (free from syntax errors) can still have poor picture quality. If the bit rate is too low, compression artifacts such as macroblock artifacts (edges) and quantization artifacts (banding) can be seen.

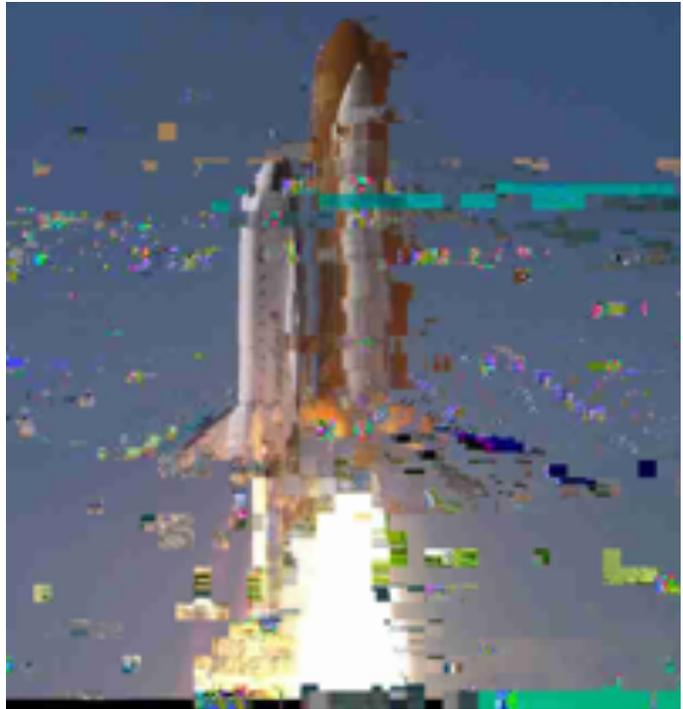


FIGURE 17. Syntax (decode) errors in the file will result in visible artifacts.



FIGURE 18. Over-compression of the video because of insufficient bit rate will cause edge and/or banding artifacts in the image.

Baseband errors in the video and audio essence in the file can be detected by decoding each frame and applying specific algorithms to the image and audio data. Video and audio dropouts will appear as solid black frames and low-level audio data (either digital zero, or below the silence threshold) respectively. Ingest errors—such as tape hits caused by a dirty head on the VTR—will have signature patterns that can be detected by QC software. Out-of-range video levels (gamut errors) or audio levels (loudness errors) can be detected by measuring the post-decode pixel and audio sample data respectively.



FIGURE 19. “Super white” regions in the image are gamut errors, indicating video levels outside the legal limits for broadcast television.

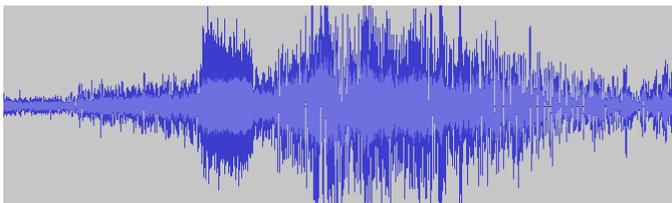


FIGURE 20. Audio levels that are too high (or too low) for extended periods in the program content will result in non-compliance with regulatory standards for program loudness.

### WORKFLOW AUTOMATION

File-based workflow processes such as transcode and QC can be automated in two ways. “Folder-based automation” (as shown in Figure 21) is used in many workflows, including large workflows that process thousands of files daily. Each software application is configured to monitor a number of “watch folders” on the media server by making periodic directory listings. When new files appear in the watch folder, they are added to the list of jobs. When a job has finished, the file can left in place or moved, depending on how the workflow is constructed. For example, a QC system might have a different output folder for files that pass and files that fail the QC check. The “quarantine” folder for failed files will be managed manually by the QC operator, who fixes (and resubmits) or rejects each file as appropriate. The output folder for files that pass QC may in turn be the input watch folder for the next application in the workflow (e.g. transcode). Basically, the QC application and the transcode application work together automatically without directly communicating with each other. Instead they use the presence of files in watch folders as an alternate method of signaling.

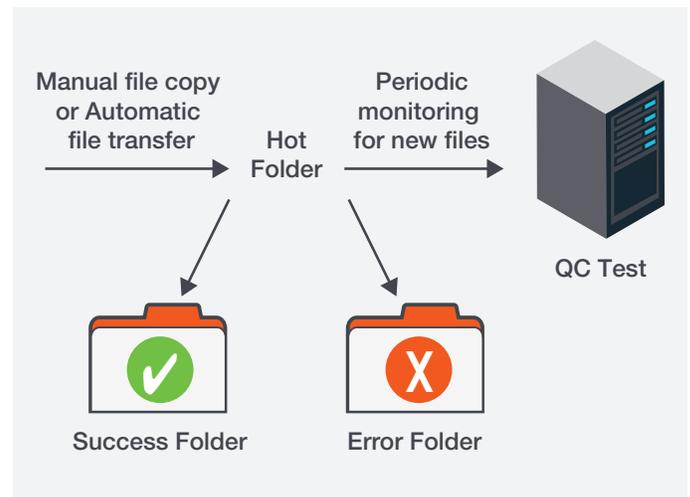


FIGURE 21. Folders on the media server can be “watched” by the automated QC system. New files are tested automatically when they appear in the folder. The files can be moved to different output folders based on the QC results.

Asset management systems typically utilize a direct control interface to each workflow application (QC, transcode, etc.) for automation. These applications are integrated with the MAM through “plug-in” software that uses the Application Programming Interface (API) of the workflow software. The Web Services architecture is commonly used for these APIs. It is so named because the standard HTTP protocol is used for request and response messages in a client-server model. Instead of a web server and a browser, the QC or transcode system is the server side and the MAM is the client side. As shown in Figure 22, the MAM always initiates a message exchange by sending a request such as “create job” or “get job status”. The server side replies with the appropriate response message. Web services protocols such as SOAP and REST use XML for the message format, making it easy to create requests and parse reply messages in software.

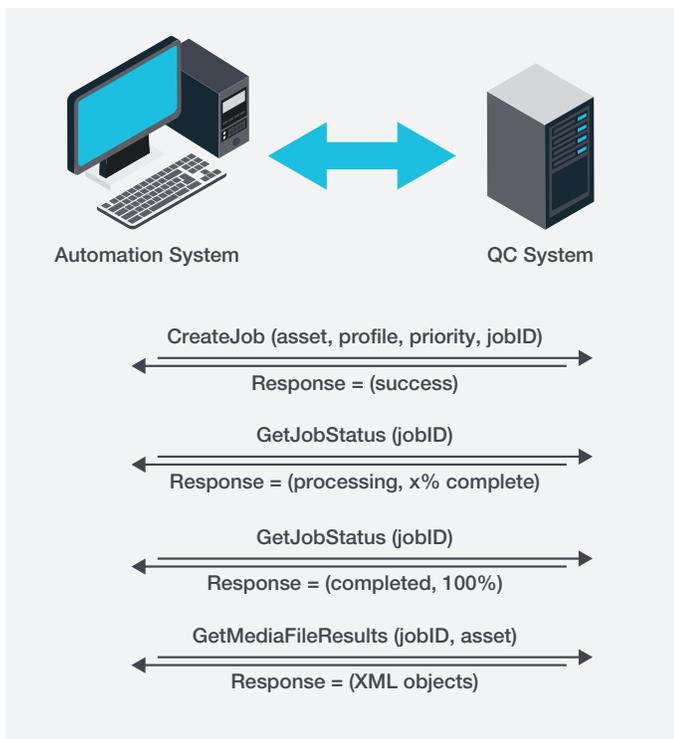


FIGURE 22. Using the API of the QC system, the automation client software can programmatically start new QC jobs, query job status, and collect the results at job completion.

With both types of automation, notification and reporting mechanisms are required. For a QC system, the operator is often notified by an email message when a file fails the QC check. That email message might contain an attachment of the QC report.

## Conclusion

The media industry is being revolutionized with the adoption of file-based workflows. Having a big picture understanding of the functions that make up file-based workflows is essential for knowing how to effectively implement quality control. These include looking at the overall physical architecture, implementing media asset management systems and managing the content from acquisition to delivery. New applications—such as streaming delivery—are now possible and improve operational efficiencies. Traditional video technologies are being adapted to work with general-purpose IT technologies. Ongoing development ensures that tomorrow’s workflows will evolve from those used today.

A key component of a file-based workflow is quality control. QC software can be used to assist the human operator, by finding both visible defects and those hidden within the file. Compliance to a broadcaster’s delivery specification can be determined prior to delivery, avoiding rejected content. Automated QC can scale in capacity to meet the growing volume of content that must be tested.

## List of Acronyms

<b>ABR</b>	Adaptive bit rate	<b>MAM</b>	Media Asset Management
<b>AMWA</b>	Advanced Media Workflow Association	<b>MP</b>	Material Package
<b>API</b>	Application programming interface	<b>MPEG</b>	Moving Picture Experts Group
<b>ARD</b>	Arbeitsgemeinschaft der öffentlich-rechtlichen Rundfunkanstalten der Bundesrepublik Deutschland	<b>MXF</b>	Material Exchange Format
<b>AS</b>	Application Specification	<b>NABA</b>	North American Broadcasters Association
<b>ASI</b>	Asynchronous Serial Interface	<b>NAS</b>	Network-attached storage
<b>ATSC</b>	Advanced Television Systems Committee	<b>NFS</b>	Network File System
<b>AVC</b>	Advanced Video Coding	<b>OP</b>	Operational Pattern
<b>CDN</b>	Content Delivery Network	<b>OPL</b>	Output Profile List
<b>CIFS</b>	Common Internet File System	<b>PAD</b>	Prêt à Diffuser
<b>CPL</b>	Composition Playlist	<b>QC</b>	Quality control
<b>CPU</b>	Central Processing Unit	<b>RAID</b>	Redundant array of independent disks
<b>DI</b>	Digital Intermediate	<b>REST</b>	Representational state transfer
<b>DPP</b>	Digital Production Partnership	<b>RGB</b>	Red, green, blue
<b>DPX</b>	Digital Picture Exchange	<b>SAN</b>	Storage area network
<b>DVB</b>	Digital Video Broadcasting	<b>SD</b>	Standard definition
<b>EBU</b>	European Broadcasting Union	<b>SDI</b>	Serial Digital Interface
<b>ENG</b>	Electronic news-gathering	<b>SMB</b>	Server Message Block
<b>FIMS</b>	Framework for Interoperable Media Services	<b>SMPTE</b>	Society of Motion Picture and Television Engineers
<b>FP</b>	File Package	<b>SOA</b>	Service-oriented architecture
<b>FTP</b>	File Transfer Protocol	<b>SOAP</b>	Simple Object Access Protocol
<b>GOP</b>	Group of Pictures	<b>SSiP</b>	Simple Studio Profile
<b>HD</b>	High definition	<b>TIFF</b>	Tagged Image File Format
<b>HTTP</b>	Hypertext Transfer Protocol	<b>TS</b>	Transport stream
<b>IMF</b>	Interoperable Master Format	<b>UHD</b>	Ultra-high definition
<b>iSCSI</b>	Internet Small Computer Systems Interface	<b>VM</b>	Virtual machine
<b>IT</b>	Information technology	<b>VOD</b>	Video on demand
<b>JPEG</b>	Joint Photographic Experts Group	<b>VTR</b>	Video tape recorder
<b>KLV</b>	Key-Length-Value	<b>XML</b>	Extensible Markup Language
<b>LAN</b>	Local Area Network	<b>ZDF</b>	Zweites Deutsches Fernsehen



## Contact Information:

**Australia\*** 1 800 709 465  
**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 / 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  
**Indonesia** 007 803 601 5249  
**Italy** 00800 2255 4835  
**Japan** 81 (3) 6714 3010  
**Luxembourg** +41 52 675 3777  
**Malaysia** 1 800 22 55835  
**Mexico, Central/South America and Caribbean** 52 (55) 56 04 50 90  
**Middle East, Asia, and North Africa** +41 52 675 3777  
**The Netherlands\*** 00800 2255 4835  
**New Zealand** 0800 800 238  
**Norway** 800 16098  
**People's Republic of China** 400 820 5835  
**Philippines** 1 800 1601 0077  
**Poland** +41 52 675 3777  
**Portugal** 80 08 12370  
**Republic of Korea** +82 2 6917 5000  
**Russia / CIS** +7 (495) 6647564  
**Singapore** 800 6011 473  
**South Africa** +41 52 675 3777  
**Spain\*** 00800 2255 4835  
**Sweden\*** 00800 2255 4835  
**Switzerland\*** 00800 2255 4835  
**Taiwan** 886 (2) 2656 6688  
**Thailand** 1 800 011 931  
**United Kingdom / Ireland\*** 00800 2255 4835  
**USA** 1 800 833 9200  
**Vietnam** 12060128

\* European toll-free number. If not accessible, call: +41 52 675 3777

Find more valuable resources at [TEK.COM](http://TEK.COM)

