LONG-TERM ASSET STORAGE ARCHIVE AND PRESERVATION WITH AXF

Nicole Jacquemin (Front Porch Digital, Metz-Tessy, France)

Many media organizations today—from broadcasters to post houses to sports teams to national archives—are still working in a hybrid content workflow that relies on legacy videotape and film-based assets alongside file-based ones. Automated solutions currently exist to rationalize the economics and technical aspects of migrating these legacy assets to digital files en masse, so media organizations do have an automated way of not only preserving those assets for the long term, but making them easier to access, manage, and market. Moving legacy assets from analog to digital and operating within a completely file-based workflow is an obvious way to improve efficiency, but once the analog-to-digital migration has happened, how can an organization ensure those assets are accessible, backed up, and protected?

The answer for many media organizations is a content storage management system.

1. What is a content storage management system?

A content storage management (CSM) system is the software abstraction layer that automatically retrieves broadcast-quality content from a data tape library (with the aid of a robot) or from a data server, delivering it to a workstation, a play-out device, or to wherever else might be needed. CSM systems were developed to help content owners cope with what would otherwise be an overwhelming volume of content, to address the video-specific complexity of that content, and to prevent content loss. All of these capabilities are critical for media companies, as content is the very lifeblood of their business.

To achieve cost-effectiveness, a typical file-based workflow's storage infrastructure is usually composed of four tiers: online, the most expensive, made up of video servers and editing systems; near-line, comprising networks and disk storage arrays; archive, composed of data or optical tape libraries; and offline, usually tapes located on physical shelves. Each provides differing content access and retrieval times, but is also characterized by significantly different capital costs. Online storage is the most expensive but provides the most immediate access to content, while offline storage is the least expensive and least accessible.

CSM middleware solutions run on one or many distributed servers, providing direct integration between the so-called media network, which connects the various devices that produce or consume file-based content, and the storage network, which connects the near-line and archive storage tiers.

The CSM solution is one of the most critical elements of the workflow. Running quietly and obediently in the background, it does the fetching and carrying so that all the other systems work up to their potential.

CSM solutions are designed from the ground up to serve demanding media-centric operations and their highly active, symmetrical nature. In terms of asset backup, CSM systems can automatically replicate file-based assets, creating duplicate copies on multiple (and portable) data tape media very rapidly and without any user intervention while sharing the same management and storage infrastructure. Those copies can remain within the system to provide online resiliency or be transported to offline storage facilities for efficient and cost-effective content protection.

2. CSM vs. HSM

Some companies still think that digital media does not require any special consideration—that data is just data—and that storing video safely and effectively is as simple as choosing a data storage technology alone. For those companies, hierarchical storage management (HSM) is a common choice. HSM systems grew up in the IT world and were designed to move files between near-line spinning disk and data tapes.

So while HSM and CSM systems share the ability to move data between storage devices or media, it is our opinion that HSM systems are not optimized for managing *digital* media as CSM systems are. CSM systems are designed to cope with the special properties and requirements of digital media files as they are moved about in the workflow of a media organization. CSM can fulfill a broad set of specialized requirements beyond what HSM can do for reliable and scalable video media storage management.

Typically, CSM solutions handle an asset from online storage through near-line, archive, offline, and back again, eliminating the need for a proprietary control layer between online and near-line storage. This is one of the most important differentiators between IT-centric HSM solutions and media-centric CSM solutions, which lie at the heart of some of the most dynamic, flexible, and scalable file-based media organizations in the world.

Unlike CSM solutions, HSM solutions *age* files that have not been accessed in some time to less expensive, higher-capacity data tape media, while presenting a transparent view of the files to users regardless of where they are stored. When a user attempts to access one of these files, the HSM solution loads the applicable data tape to migrate the contents of the file back to disk storage and onward to the calling application. Perhaps the only noticeable side effect is that the file might take a little longer to open. It is also important to note that HSM systems are different than backup solutions. HSM can be seen as simple disk capacity extenders, while backup solutions ensure the ability to recover and reconstruct important data fully. This process often requires different middleware solutions to provide these functions, although occasionally the physical storage resources can be partitioned and shared between them to save capital costs.

Another distinction is that CSM systems have content awareness, which enables them to handle content as objects as opposed to simple files moving through a storage infrastructure. That is, they can group related media assets, such as a video file and associated multi-language audio tracks, as a single managed object that can be stored and retrieved as one. Media assets are typically composed of a complex collection of media and ancillary files that must be maintained carefully in order to reconstruct, access, or reuse them. This concept is usually referred to as an *object store* and is the fundamental basis of advanced CSM solutions. Rather than only maintaining a simple collection of unrelated files, paths, and folders (as in the case of HSM, a simple file system, or other technology such as LTFS), CSM solutions treat each media asset as a single unified object. The complexity surrounding the storage, recall, replication, repurposing, and transformation of these complex media objects is handled by the CSM system. Management of actual stored content, as opposed to stub files, enhances control without increasing the complexity of the storage infrastructure.

A CSM system's content awareness yields other benefits as well. Traditionally referred to as *archive software*, today's CSM solutions have evolved. Content awareness enables various media-centric features in addition to basic storage (store and restore) functionality—features such as distributed transcoding, metadata mining, file-based subjective quality analysis, timecode-based partial restore, and more—all operating in a file-based domain. Today's CSM systems also provide universal accessibility to these features via content lifecycle and policy engines, workflow tools, and open APIs for direct third-party control, integration, and collaboration.

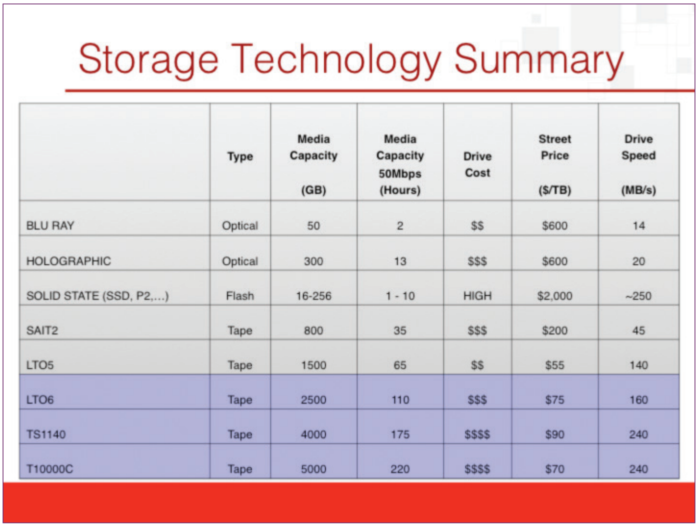
Due to the challenges of handling large digital media content, CSM solutions typically reside in the "back office" or equipment room within the media organization. They integrate directly through high-speed networks to editing systems, playout servers, newsroom systems, etc., and via the aforementioned open APIs to user-facing tools such as media asset management (MAM), broadcast automation, and business systems. These CSM systems abstract the arduous work involved in dealing with valuable high-resolution assets and often permit ubiquitous and federated access through user-friendly Web-based interfaces, shortening learning curves and allowing creative people to focus on their art in a collaborative environment rather than on technical complexities.

For example, a properly implemented CSM system enables a news producer to review a comprehensive index of archival material directly from a workstation. The index includes thumbnails and browsable proxy copies of clips in addition to metadata records to assist in identifying the best shots. Once key shots are identified, the CSM solution can perform timecode-based partial restore operations on the high-resolution versions of the content and push these segments directly to the nonlinear editing environment for finishing. This not only saves time, it gives the producer access to more content, all of which potentially improves the quality of the work. Further, because this screening is done via a Web browser directly from the producer's desktop, screening rooms and editing stations are not tied up for videotape review and shot re-ingest. Once complete, the newly produced news story can be played immediately to air via traditional methods, while, in parallel, sent back to the CSM solution, which can automatically transcode it to myriad formats and deliver it to online portals (e.g., news website, iTunes, or YouTube), driving additional viewership and potentially additional revenue.

3. Storage in a CSM environment

CSM touches many aspects of the video workflow, and accessibility of the files is of utmost importance. The way the assets are stored plays a key role in the organization's ability to access the material today and in the future.

There are many storage technologies available, with a range of capacities, transfer throughput, and price to suit different operational scenarios and budgets (Figure 1).



Storage Technology Summary

	Type	Media Capacity (GB)	Media Capacity 50Mbps (Hours)	Drive Cost	Street Price ($/TB)	Drive Speed (MB/s)
BLU RAY	Optical	50	2	$$	$600	14
HOLOGRAPHIC	Optical	300	13	$$$	$600	20
SOLID STATE (SSD, P2,...)	Flash	16-256	1 - 10	HIGH	$2,000	~250
SAIT2	Tape	800	35	$$$	$200	45
LTO5	Tape	1500	65	$$	$55	140
LTO6	Tape	2500	110	$$$	$75	160
TS1140	Tape	4000	175	$$$$	$90	240
T10000C	Tape	5000	220	$$$$	$70	240

Figure 1. Storage Technology Summary

No matter which storage technology is used, it should meet some important criteria for preserving today's more complex media assets—assets that require advanced storage methods related to the object store concept mentioned earlier. The proliferation of technology has resulted in a multitude of formats and systems for storing digital media, and often those formats and systems are not compatible with one another. Here we are not talking about interoperability of the media files themselves, but rather the actual operating system, file system, storage technology, and devices used to capture, store, and protect these media assets now and into the future. This diversity and potential long-term incompatibility makes reliable and guaranteed access to these assets complicated, expensive, and sometimes impossible. Solving the problem means establishing a common format for digital media storage that works not only with any existing system, but also systems that have yet to evolve—an open standard for the long-term storage and preservation of media assets.

Although this may seem unnecessary on the surface, there are many documented cases today where important files stored on dated technology using non-standardized methods have become inaccessible and are therefore lost forever. We may be able to recreate an MPEG-2 software decoder on whatever platforms exist 100 years from now, but are we certain we will be able to find a system compatible with FAT32 to be able to recover the MPEG-2 content itself?

So, to tackle this daunting problem, the ideal storage format should:

- Ensure long-term accessibility
- Enable self-descriptive assets and self-descriptive storage media
- Have Open Archival Information System (OAIS) preservation features (e.g., fixity and provenance)
- Encapsulate files to wrap related metadata and other files
- Be scalable for any number of elements of any size and type
- Be standardized regardless of storage media technology
- Facilitate transportability and compatibility among systems

4. Some imperfect options

4.1. Tape ARchive Format (TAR)

Tape ARchive format (TAR) has been around for decades. Initially created in the early days of UNIX to write data to sequential I/O devices for tape backup purposes, TAR is now commonly used to collect many files into one larger file for distribution or archiving, while preserving file system information such as user and group permissions, dates, and directory structures. Even though it is still used heavily today, many of its design features are considered dated. For one thing, despite following established standards, there is no true universal TAR implementation. TAR is also a legacy format that does not allow intelligent functions such as partial file restore. In addition, it does not maintain an on-media catalog of stored content, it has no resiliency feature, and it has no apparent long-term preservation features as defined in the OAIS model.

4.2. Linear Tape File System (LTFS)

The Linear Tape File System (LTFS) is a simple file system for linear data tape that makes data tapes appear as "removable storage." It is our opinion that LTFS is very useful as a physical file-based transport mechanism but not for long-term storage or preservation. It offers no media encapsulation and relies on simple folder hierarchies to form important asset relationships. It has no support for spanning across storage media, which limits file collection sizes and scalability, and it is not applicable to every storage technology.

5. Archive eXchange Format (AXF)

The Archive eXchange Format (AXF), on a path to standardization by the Society of Motion Picture and Television Engineers (SMPTE), takes the concept of the object store to a physical level by offering a self-describing, self-contained encapsulation format for complex file collections. It is an open, standardized way of storing files or file collections of any type and size—along with their associated metadata collections—on any type of storage technology or device (e.g., flash media, spinning disk, data tape, or the cloud) while remaining independent of the host operating or file system, which supports the content's long-term availability no matter how storage or file system technology evolves. AXF has preservation at its heart, including core archival characteristics such as fixity, provenance, and context, all of which are described in the well-known OAIS reference model.

5.1. What is AXF?

At the most basic level, AXF is an IT-centric file container that can encapsulate any number and any type of files in a self-contained, self-describing, protected object package. The encapsulated package contains its own file system, which abstracts the underlying operating system, storage technology, and the original file system from the AXF object and its payload. It is like a file system within a file that can store any type of data on any type of storage media (Figure 2).

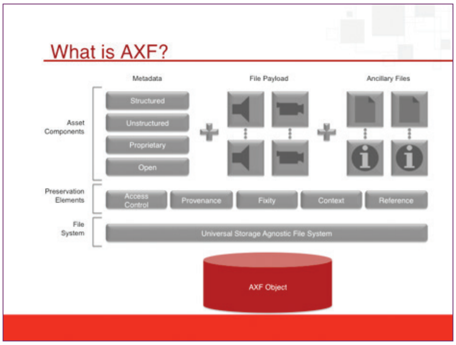

What is AXF?

Figure 2. Archive eXchange Format (AXF) Universal Object Storage Format

It supports the inclusion of any amount of open or vendor-specific, structured or unstructured metadata encapsulated as part of the object itself, strengthening its self-descriptive nature. AXF also extends its self-descriptive nature to the storage media that contain AXF objects, allowing access using any AXF-aware system, and supporting long-term accessibility and protection regardless of whether the original system is available or not.

On the surface, one can say that AXF provides the same functionality as LTFS in terms of accessibility, ease of use, and portability, but then layers on top of it universality and the long-term storage and preservation features required in archival applications. AXF can be seen as the object-store extension of LTFS, overcoming many of its shortcomings such as lack of media-spanning support, no file encapsulation, and no metadata support.

5.2. The embedded file system

The embedded file system approach is a key attribute of AXF. It allows AXF to be both content- and storage-agnostic. In other words, because the AXF object itself contains the file system, it can exist on any generation of data tape, spinning disk, flash, optical media, or other storage technology.

Because of this neutrality, AXF supports the modern generation of data tape technologies—such as LTO5, TS1140, and T10000C—and because there is no dependency on the features of the storage technology itself, it supports legacy storage formats as well.

5.3. Comparison of AFX to other storage formats and approaches

It is our opinion that AXF offers significant advantages over other formats and approaches such as TAR and LTFS for long-term storage, protection, and preservation, including:

- AXF can scale without limit, which distinguishes it from legacy container formats such as TAR. Like AXF, TAR uses a file container approach that works on any file type of any individual or total file size with support for multiple operating systems. However, TAR's age and tape-based roots yield limitations. For example, it incorporates neither descriptive metadata support nor a central index for file payload information, which makes random access to files challenging and slow. In large TAR archives, the performance penalty is significant, effectively making the format unsuitable for a situation where random access to individual files is required. Certainly, TAR has evolved over the decades, but typically in divergent paths that lead away from its open-source origins. As a result, it is difficult or impossible to recover some TAR packages today.

- Also in contrast to TAR, AXF incorporates resiliency features that make it possible to recover object contents, descriptive metadata, and media catalogs in many failure and corruption situations. Also unlike TAR, AXF incorporates fixity and error-checking capabilities in the form of multiple per-file and per-structure checksums.

- The embedded file system enables AXF to translate between any generic set of files and logical block positions on any storage medium, whether the medium has its own file system or not. This abstracts the underlying file system and storage technology, allowing systems that comprehend AXF to ignore any of their complexities and limitations.

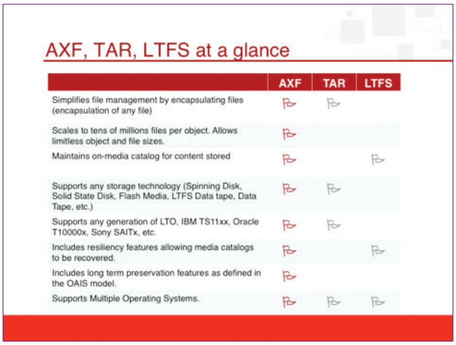
- While AXF can work in harmony with LTFS, it also has advantages over it. LTFS relies on storage technology elements—such as partitioning and file marks on data tape—which hinders its storage capabilities and its performance. Likewise, LTFS is ineffective for complex file collections containing tens of thousands or even millions of related elements, as it lacks any form of encapsulation but instead relies on file and path arrangements.

- AXF can support any number and type of files in a single encapsulated package, which means these AXF objects can grow exponentially in size. With its support for spanning objects across media (such as over multiple data tapess), AXF has significant advantages over LTFS, which offers no spanning support and is therefore ineffective in large-scale archives typical in media operations.

- For the preservationist community, AXF offers support for the core OAIS reference model, with built-in features such as fixity (per-file checksums and per-structure checksums), provenance, context, reference, open metadata encapsulation, and access control.

- Once content is stored in AXF, the media itself can be transported directly to any system that also comprehends AXF, offering the same "transport" capabilities of LTFS with the additional features highlighted above.

These factors are key to AXF's ability to support large-scale archive and preservation systems as well as simple, stand-alone applications (Figure 3).



AXF, TAR, LTFS at a glance

	AXF	TAR	LTFS
Simplifies file management by encapsulating files (encapsulation of any file)	⚐	⚐	
Scales to tens of millions files per object. Allows limitless object and file sizes.	⚐		
Maintains on-media catalog for content stored	⚐		⚐
Supports any storage technology (Spinning Disk, Solid State Disk, Flash Media, LTFS Data tape, Data Tape, etc.)	⚐	⚐	
Supports any generation of LTO, IBM TS11xx, Oracle T10000x, Sony SAITx, etc.	⚐	⚐	
Includes resiliency features allowing media catalogs to be recovered.	⚐		⚐
Includes long term preservation features as defined in the OAIS model.	⚐		
Supports Multiple Operating Systems.	⚐	⚐	⚐

Figure 3. AXF Advantages Over TAR and LTFS

5.4. How does AXF work?

AXF is designed so that each AXF Object (or package) has three main components regardless of what technology is used to store them (e.g., spinning disk, flash media, data tape without a file system, or data tape with a file system). These are:

- Each AXF Object originates with an AXF Object Header—a structure containing descriptive metadata such as the AXF Object's unique identifier (UUID and UMID), creation date, object provenance, and file-tree information including file permissions, and paths. Following the AXF Object Header is any number of optional AXF Generic Metadata packages. The AXF Generic Metadata Packages are self-contained, open metadata containers in which applications can include AXF Object-specific metadata. This metadata can be structured or unstructured, open or vendor-specific, binary, or XML.

- The next part of the AXF Object construct is the AXF File Payload—the actual byte data of the files encapsulated in the object. The payload consists of any number of triplets—File Data + File Padding + File Footer. File padding, which ensures alignment of all AXF Object elements on storage medium block boundaries, is key to the AXF specification. The File Footer structure contains the exact size of the preceding file, along with an optional file-level checksum designed to be processed on the fly by the application during restore operations with little or no overhead.

- The final portion of an AXF Object is the AXF Object Footer, which repeats the information contained in the AXF Object Header and adds information captured during the AXF Object's creation, including per-file checksums and precise file and structure block positions. The AXF Object Footer is important to the resiliency of the AXF specification because it allows efficient re-indexing by foreign systems when the media content is not previously known, enabling media transport between systems that follow the AXF specification.

Because of this standardized approach to the AXF Object construct (Figure 4), which abstracts the underlying complexities of the storage media itself, access to the content is supported regardless of the evolution of technology now and into the future.

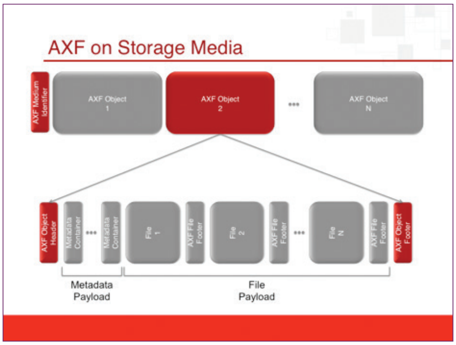

AXF on Storage Media

Figure 4. The AXF Container

5.5. Special structures for use with linear data tape

When used with the linear data tape typical in large-scale archives today, an AXF implementation includes three additional structures to incorporate key self-describing characteristics on the medium itself, ensuring recoverability and transportability:

- The first structure, which appears on the medium, is an ISO/ANSI standard VOL1 volume label. This is included for compatibility purposes with legacy applications to ensure they do not erroneously handle AXF formatted media and to signal applications that do understand AXF that they can proceed to access the objects contained on the medium.

- The second structure is the Medium Identifier, which contains the AXF volume signature and other information about the storage medium itself. The implementation of the Medium Identifier differs slightly depending on whether the storage medium is linear or nonlinear, and whether it includes a file system or not, but the overall structures are fully compatible.

- The third structure is the AXF Object Index, which is an optional structure that assists in the recoverability of AXF-formatted media. Information contained in this structure is sufficient to recover and reconstruct the entire catalog of AXF Objects on the storage medium. In a case where the application has not maintained the optional AXF Object Index structures, the contents of each AXF Object can still be reconstructed by processing each AXF Object Footer structure, adding to the resiliency of the format.

5.6. Who can use AXF?

Anyone. AXF was developed to meet a broad spectrum of user needs—from accessing petabytes of data in a high-performance environment to simply encapsulating a few files and sending them to a friend via email. AXF is scalable to accommodate an operation of any size or complexity. In all cases, AXF offers an abstraction layer that hides the complexities of the storage technology from the higher-level applications, while it also offers encapsulation, provenance, fixity, portability, and preservation characteristics. In addition, the same self-describing AXF format can be used interchangeably on all current storage technologies, such as spinning disk, flash media, and data tape from any manufacturer.

5.7. Where does AXF stand now?

Work is underway within SMPTE and its AXF Working Group to standardize AXF and promote it as an industry-wide method for storage and long-term preservation of media assets. Further, the committee hopes its work will extend far outside of the media and entertainment space and into the broader IT community because of its wide reaching applicability.

In April 2013, the working group submitted its final draft of the AXF specification for two-week review, which is the final step before balloting to become an official SMPTE standard. In September 2013, the committee agreed that all comments from the review have been adequately addressed. As of this writing, a final draft of the specification was being prepared for the official SMPTE ballot. After a two-week balloting period, the AXF format will become an official SMPTE standard.

5.8. The bottom line

AXF has the ability to support interoperability among systems, help ensure long-term accessibility to valued assets, and keep up with evolving storage technologies. It offers many present and future benefits for any enterprise that uses media—from heritage institutions, to schools, to broadcasters, to simple IT-based operations—and is well on its way to becoming a world-wide, open standard for file-based archiving, preservation, and exchange.

More information on AXF and standards-body activities is available at OpenAXF.org and smpte.org.