SMPTE ST 2034-1:2017
Revision of
SMPTE ST 2034-1:2014

SMPTE STANDARD

Archive eXchange Format (AXF) — Part 1: Structure & Semantics

Table of Contents

Approved
February 27, 2017

Foreword

SMPTE (the Society of Motion Picture and Television Engineers) is an internationally-recognized standards developing organization. Headquartered and incorporated in the United States of America, SMPTE has members in over 80 countries on six continents. SMPTE's Engineering Documents, including Standards, Recommended Practices, and Engineering Guidelines, are prepared by SMPTE's Technology Committees. Participation in these Committees is open to all with a bona fide interest in their work. SMPTE cooperates closely with other standards-developing organizations, including ISO, IEC and ITU.

SMPTE Engineering Documents are drafted in accordance with the rules given in its Standards Operations Manual.

SMPTE ST 2034-1 was prepared by Technology Committee 31FS on File Formats and Systems.

Intellectual Property

At the time of publication, no notice had been received by SMPTE claiming patent rights essential to the implementation of this Engineering Document. However, attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. SMPTE shall not be held responsible for identifying any or all such patent rights.

Introduction

This section is entirely informative and does not form an integral part of this Engineering Document.

The Archive eXchange Format (AXF) is an open format that supports interoperability among disparate data storage systems and ensures long–term availability of data, no matter how storage or file system technologies evolve. AXF inherently supports interoperability between existing, discrete storage systems, irrespective of the operating and file systems used, and also future–proofs digital storage by abstracting the underlying technology so that content remains available across generations of technology development.

At the most basic level, AXF is a file container that can encapsulate any number, size, and type of files in a fully self–contained and self–describing package. The package contains its own light-weight file system, which establishes independence from underlying operating systems, storage technologies, and file systems and can store any type of data on any type of storage media. Inside its packaging, AXF can contain metadata of any format, applicable to either AXF Objects or to individual files contained within AXF Objects; AXF also carries key preservation information, such as provenance, fixity, and the like — all key to ensuring long-term robustness and recoverability.

Historically, digital archive systems have used media data storage formats that are proprietary to their manufacturers, either intentionally or due to the lack of established standards. There have been neither interchange of media nor interoperability of archive systems between manufacturers and in some cases between different archive systems from the same manufacturer. Archives could be orphaned due to support ending for the systems used to create data archives. End users and manufacturers recognized that the proprietary nature of archive systems and the data stores that they create result in significant costs of operation that are unnecessary. These costs could be avoided if there were standardization of the format used for storage of the data on media and for transfer of the data between systems and locations. AXF permits separating the stored content from the systems that create and recover sets of data, thereby enabling refreshing of storage technology, recovering sets of data that otherwise would have been orphaned, and transferring sets of data between systems and locations.

This standard specifies a structure for data that can be written to any current or future data storage subsystem, regardless of the type of media on which it is stored. The data can include any types of files and associated metadata that are stored and transferred together in a structure called an "AXF Object." A single AXF Object can be spanned across multiple physical media, can be copied from one set of physical media to

another, and is agnostic to the Storage Media Type on which it is stored, e.g., spinning disc or linear tape. Regardless of the Storage Media Types on which they are stored, AXF Objects are identically structured and formatted for any given set and relationship of contained files and metadata.

AXF initially arose from the storage needs of the audiovisual production and archiving communities but quickly encompassed any type of file-based data. The transition to file-based workflows led to a new set of requirements throughout pre-production, production, distribution, storage, and preservation processes. Those requirements included long-term archiving of finished and unfinished materials, writing data to any type of storage subsystem using a standard scheme, transporting formatted archives between systems and locations using either media or networks, and allowing extensibility sufficient to accommodate any type of file, of any size, from any source, as well as adoption of any future storage technologies. AXF was created to address these requirements.

Audiovisual content archiving spans a wide range of content and data archiving systems and practices. At the time this standard was written, many different methods and media were commonly used to store file-based audiovisual content and its supporting information. Examples range from individual hard drives, solid state drives, and linear magnetic tape drives in small organizations to large spinning disc arrays in combination with very large robotic systems with multiple robots, each having multiple drives, in very large cultural, scientific, and legal archives. Applications in other industries that could benefit from the methods defined herein include medical imaging, geophysical exploration, scientific research, and similar high-volume producers of data.

The cultural, scientific, and business value of assets stored on these data systems is significant. Methods for storage, interchange, transport, and preservation of such assets, both locally and remotely, over both short and very long retention periods, demands a standardized, well-documented, non-manufacturer-specific method of writing data to any data storage system, from which the data then can be recovered and its contents used, updated, or transferred to another data storage system. All that would be necessary to achieve these objectives is a mechanism for recovering data from the media on which it is stored, plus utilities or applications that implement AXF.

The AXF standard creates a common method of writing individual files or related sets of files, and relevant metadata, onto data storage subsystems so that the structure of an AXF Object will remain the same no matter what vendor equipment or Storage Media Type is used. As long as the media remains viable and data can be read from that media, it will be possible to recover an AXF Object and unwrap its contents with a suitable utility or application running on whatever platform is current at the time. The AXF Object also has to be able to be recovered and stored on future data storage systems without requiring any changes to its contents simply to accomplish the act of medium migration, but it also needs to allow changes to its contents, in case updating is needed to data that already has been archived.

AXF addresses these needs through a combination of predefined eXtensible Markup Language (XML) schema fields, defined binary data structures that enable an AXF Object to carry any type of file within its File Payload, internal file system functionality, and key metadata enabling the spanning of AXF Objects across multiple physical media. The XML schema also enables essential information about an AXF Object and its contents to be read without having to process all the information within the AXF Object.

In addition to media interchange, AXF enables the interoperability of disparate systems through networks because it is structured as a streaming data set. Such interconnections enable seamless movement of AXF Objects from systems that create them, to systems that do not recognize the AXF protocol but store the AXF Object files nonetheless (perhaps in “cloud” storage), then to systems that are designed to recover data from AXF Objects.

Functionally, AXF acts like a file wrapper or a repository for all types of data without constraint. Unlike media-centric file formats such as MXF, which are similar in that they wrap essences, AXF can contain any number or types of files of any size encapsulated in an AXF Object. It is applicable across a much broader variety of file storage user groups than any media-specific file wrapper. Types of data can include media essence files, related metadata files, production files (such as word processing documents, hypertext documents, associated essence, applications, spreadsheets, and database copies), or any other type of data that users wish to store together. Unlike other file wrapper definitions, it is payload agnostic and does not require any special mappings or adaptations to accept the data an AXF Object carries.

AXF accommodates very large file sizes and quantities within AXF Objects. In the current version of this standard, 64-bit numbers are used to define the sizes of various parameters applicable to elements of AXF Objects. 64-bit numbers can express values up to 18.44674 x 10^{18} (e.g., 18.44674 petabytes). Use of 64-bit numbers thus can define file sizes in bytes, numbers of files, numbers of media in a spanned set, and similar characteristics up to 18.44674 x 10^{18} of any particular element. If future requirements exceed the number spaces provided in this document, there is nothing fundamental that limits any particular parameter to expression using a 64-bit number. Future revisions of this standard could adopt larger number spaces (e.g., 96-bit, 128-bit, etc.) for those parameters requiring them. The net result is effectively unlimited storage capability within AXF Objects, in terms of file sizes, numbers of files in an AXF Object, number of AXF Objects on a medium, number of media in a spanned set, and the like.

AXF enables updating AXF Objects when additions of, modifications to, or deletions of files or information that they contain are needed. The functionality to modify AXF Objects is provided by linking "Supplemental" AXF Objects, written into an archive system at a later time to an original ("Anchor") AXF Object. A Supplemental AXF Object updates contents of previous AXF Objects without requiring the original AXF Object itself to be modified. Since the original content of the AXF Object is retained in its original form, it is possible to restore either the original or the modified version whenever necessary. Additional Supplemental AXF Objects can be added in a chain, with restoration of the current or any earlier version possible at any time. When AXF Objects are refreshed by copying them to new media, it is possible to consolidate an Anchor Object and its Supplemental Object(s) into a single, new AXF Object. In doing so, it is possible to retain all of the constituent Objects of the Collected Set to which they belong, so that all earlier versions still can be reconstituted in the future.

AXF abstracts the storage of data from the applications that create AXF Objects and from the operating systems, file systems, drivers, and drives that store data on media. By this mechanism, any of the surrounding hardware and software components of systems can be replaced without affecting the data and its formatting within AXF Objects. A simplified view of where AXF fits into a basic stack is shown in Figure 1.

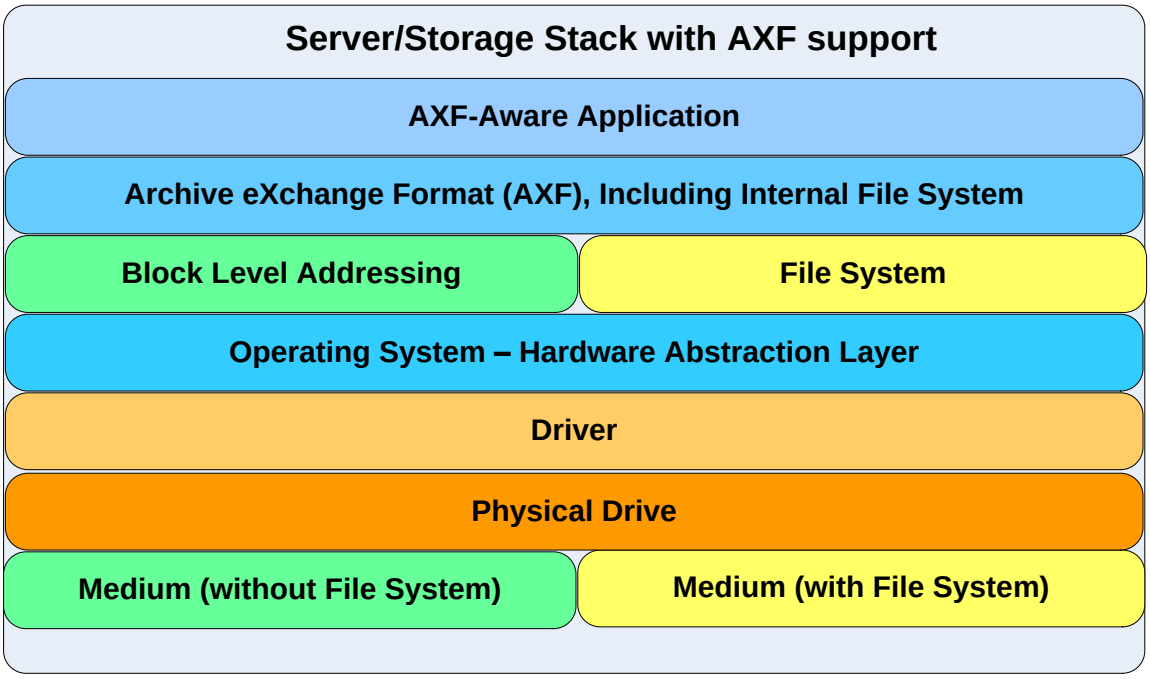

Figure 1 – Hardware/Software Stack Incorporating AXF Writing to and Reading from Media

AXF is designed so that each AXF Object comprises four main components, regardless of the technology that is used to store it. These components are:

Object Header – Each AXF Object begins with an Object Header, which contains descriptive XML metadata such as a unique identifier (UUID) for the AXF Object, information regarding its origin, its creation date, and a full index of all the files and folders contained in the Object, including file permissions and the like.

Generic Metadata Containers – Following an Object Header can be any number of optional Generic Metadata Containers. Such containers are self–contained, open metadata containers in which applications can place AXF-Object-specific metadata that is not part of the AXF Object File Payload. The metadata can be structured or unstructured, open or vendor-specific, binary, XML, plain text, or any other format.

AXF Object File Payload – Following any Generic Metadata Containers is the AXF Object File Payload. It contains the files encapsulated in the AXF Object. The File Payload consists of any number of triplets: File Data + File Padding + File Footer. File Padding ensures alignment of all AXF Object elements on the boundaries of Chunks into which each AXF Object is divided, thereby enabling addressing, by location within the AXF Object, by its internal file system. File Footer structures contain full information about the preceding file, along with a file-level checksum designed to be processed on-the-fly, with little or no overhead, during restore operations by an application. The information in File Footers enhances the resilience of AXF, as it can be used to recover File Payload data even if Object Header and Footer structures are missing or corrupt.

Object Footer – Completing an AXF Object is an Object Footer. It repeats the information contained in the Object Header and adds information captured during creation of the AXF Object, including per-file checksums, precise file sizes, and file positions within the AXF Object. The Object Footer is important to the interchange of an AXF Object because it allows efficient indexing by foreign systems when the media content is not previously known, thereby enabling media transport between systems that follow the AXF standard. It is one of the key structures that support the self-describing nature of AXF.

Other significant structures in the AXF protocol are AXF Medium Identifiers and AXF Object Indices. AXF Medium Identifiers are used on media to indicate formatting of the media according to the AXF protocol and to provide unique identification of the media. AXF Object Indices are optional compilations of the information in all Object Footers preceding each AXF Object Index on a medium, providing a single structure from which it is possible to obtain complete information on the contents of the preceding portion of a medium. When an AXF Object Index is the last structure on a medium, complete information about all AXF Objects stored on the medium can be obtained efficiently in one place.

AXF does not require a system to be fully compliant with this standard for it to be able to use and store AXF-generated AXF Objects. The initial adoption of AXF is anticipated to be in applications that create AXF Objects that then are stored on non-AXF-aware storage systems. Because the AXF Objects do not require a storage system to know that the AXF Objects are AXF-formatted, the AXF Objects will be viewed simply as files to be stored and retrieved. All that will be necessary to read AXF Objects will be the software and hardware needed to read the physical storage medium. As adoption grows, files can be moved into and out of AXF-aware systems as necessary, with the full range of features becoming available on systems that are AXF compliant. AXF-compliant applications will be able to read stored AXF Objects from any current operating system without unpacking entire AXF Objects to see critical metadata. Moreover, Archives, or AXF Objects within archives, also can span different types of media, allowing for flexibility within mixed-media archives and for AXF Objects to be identical, regardless of the media on which they are stored.

AXF offers resilience to data corruption and loss. AXF Object Indices, repeated identifier instances, and cryptographic hash checksums on both contents and AXF Objects allow for data corruption to be identified and mitigated. Even in catastrophic events, such as the loss of an external database containing records of the contents of an archive, the content database can be recreated by reading the archive and regenerating the archive-wide database from the records within the AXF Objects. This standard also enables the addition of more powerful data corruption recovery methods in future revisions by including provisions for incorporating forward error correction codes. AXF provides key features to identify and treat data corruption and loss across all types of storage formats. It allows big data and small to be processed in a consistent and standardized way.

1 Scope

This standard is Part 1 of a series of documents that specify a general-purpose format for the storage and/or communication of information in bulk form. The format is named the Archive eXchange Format (AXF). The format described is intended both for interchange between systems and to serve as a native format within systems.

This standard identifies two major categories of data storage media and specifies the basic structures of data stored on those Storage Media Types. It specifies a number of structural elements for use in constructing the appropriate structures for use on each of the Storage Media Types. It defines the semantics of data contained within fields specified for use in the structural elements. The structural elements themselves are documents coded in the eXtensible Markup Language (XML), and this document defines an XML Schema Description (XSD) file for use in formulating the XML documents to be used for the structural elements of AXF Objects.

2 Conformance Notation

Normative text is text that describes elements of the design that are indispensable or text that contains the conformance language keywords: "shall," "should," or "may." Informative text is text that is potentially helpful to the user, but not indispensable, and that can be removed, changed, or added editorially without affecting interoperability. Informative text does not contain any conformance keywords.

All text in this document is, by default, normative, except: the Introduction, any section explicitly labeled as "Informative," or individual paragraphs that start with "Note:".

The keywords "shall" and "shall not" indicate requirements strictly to be followed in order to conform to the document and from which no deviation is permitted.

The keywords, "should" and "should not" indicate that, among several possibilities, one is recommended as particularly suitable, without mentioning or excluding others; or that a certain course of action is preferred but not necessarily required; or that (in the negative form) a certain possibility or course of action is deprecated but not prohibited.

The keywords "may" and "need not" indicate courses of action permissible within the limits of the document.

The keyword “reserved” indicates a provision that is not defined at this time, shall not be used, and may be defined in the future. The keyword “forbidden” indicates “reserved” and, in addition, indicates that the provision never will be defined in the future.

A conformant implementation according to this document is one that includes all mandatory provisions ("shall") and, if implemented, all recommended provisions ("should") as described. A conformant implementation need not implement optional provisions ("may") and need not implement them as described.

Unless otherwise specified, the order of precedence of the types of normative information in this document shall be as follows: Normative prose shall be the authoritative definition; tables shall be next, followed by formal languages, then figures, and then any other language forms.

3 Normative References

The following standards contain provisions that, through reference in this text, constitute provisions of this standard. At the time of publication, the editions indicated were valid. All standards are subject to revision, and parties to agreements based on this standard are encouraged to investigate the possibility of applying the most recent editions of the standards indicated below.