AAF

An industry-driven open standard for multimedia authoring

Wouldn't it be great to be able to share audio, video, paint, and 3D files easily between editing and effects applications, as well as the data that holds them all together? To be able to import not just the media, but all the creative decisions that turn raw sound and imagery into finished content?

This document is intended for developers with an interest in the Advanced Authoring Format. It describes AAF's origins, discusses the need for such an industry-wide standard, and provides an overview of the basic concepts.

Contents

The Advanced Authoring Format (AAF) is a multimedia file format that enables content creators to easily exchange digital media and metadata across platforms, and between applications. The AAF simplifies project management, saves time, and preserves valuable metadata that was often lost when transferring media between applications in the past.

AAF
Advanced Authoring Format

AES
Audio Engineering Society

ASF
Advanced Streaming Format

AUID
Advanced Unique Identifier

EBU
European Broadcasting Union

KLV
Key Length Value

MXF
Media Exchange Format

NCITS
National Committee for Information Technology Standards

OMFI
Open Media Framework Interchange

SMPTE
Society of Motion Picture and Television Engineers

SSF
Structured Storage Format

UMID
Unique Material Identifier

How does AAF relate to SMPTE?

AAF is a software implementation of SMPTE metadata and SMPTE labels, designed particularly to make it easy to work with large collections of interrelated sets of metadata and essence. SMPTE "KLV" is at its heart. Besides the ability to format and manipulate metadata itself, the AAF software toolkit provides added capabilities for management of metadata sets, user extensions, and pluggable modules.

AAF is moving through committee in SMPTE. AES will consider AAF for the object-oriented description of projects. Some elements of AAF have been incorporated into MPEG-4, and MPEG-7 is harmonizing metadata with SMPTE. The Pro-MPEG forum is studying AAF compatibility.

Background

High-end, rich content authoring is a delicate struggle—bringing together highly disparate source media, and arranging these elements to form a coherent whole.

Consider the example of putting together all of the audio elements for a film soundtrack. This involves transferring all of the music tracks, the ambient sound tracks, the performer's synch sound and ADR, and the Foley effects from their original source, remixing or editing all of them, and doing frame-accurate synchronizations to the motion picture elements. This process requires a lot of information about each audio source element, as well as information about other media associated with it during playback.

The media industry uses a wide range of source materials, as well as a set of highly varied capture tools with very different constraints (cameras, keyboards, audio input sources, scanners, disk drives). This leads to a great deal of time and effort spent converting media into formats that can be used by authoring applications.

While both SMPTE and EBU have effectively addressed conversion problems in the dedicated hardware world by creating a set of standards, the same cannot yet be said for software. Working with EBU/SMPTE, the AAF Association is spearheading an effort to make the Advanced Authoring Format such a software standard.

The EBU/SMPTE Task Force

In 1996, a joint task force of the SMPTE and the European Broadcasting Union (EBU) was formed to look into the challenges, roadblocks, and standards development opportunities available in the looming shift in worldwide broadcasting from analog to digital. Thus was born the EBU/SMPTE **Task Force for Harmonized Standards for the Exchange of Program Material as Bit Streams**.

An advanced summary of findings was given at NAB in April 1998, with a fully finalized report delivered to an enthusiastic audience of over 300 at IBC in Amsterdam in September 1999. The final report was published in the September Journal, one of the largest and most widely distributed issues ever.

This was truly a Herculean effort involving the time and energy of literally hundreds of volunteers and staff who met 17 times since the project was begun two years earlier. As Engineering Vice-President Bill Miller wrote in the Journal introduction to the report, "This joint effort of SMPTE and the European Broadcasting Union (EBU) is, we believe, one of the most significant achievements in the history of the two organizations."

SMPTE Abstract

" While digital video can be stored in an archive as program content, there is no uniform way to describe relationships between the video itself and metadata that describes how this video is to be displayed, edited, or related to other program material. AAF resolves this dilemma by providing a uniform way for content creators to link information about content to the content itself. "

AAF Association

" CNN/Turner Broadcasting needs standards to describe complex media and enable interchange between different systems. We look to the AAF Association to move these efforts forward. Without this work, vendors are faced with developing custom integration that is costly, slow, and not extensible."

Gordon Castle, CNN

A Standard Format for File Exchange

The sheer number of available digital media file formats (AVI, AIFF, DV, TIFF, etc.), each with its own strength or specific quality (e.g. preferred compression codec, optimized file size, preferred color resolution, or operating system platform), makes it necessary to perform many file format conversions in the course of producing a high-quality end product.

The Advanced Authoring Format helps the content creation and authoring process by addressing the compatibility between formats. In this way, AAF allows users to apply their creative energies to the quality of the media *compositions,* relieving them from having to deal with unnecessary and painful interchange issues. It also allows software development to focus on improvements to the authoring application's feature set.

Universal Digital Media Authoring

Multimedia authoring applications read and manipulate certain types of media, and save the resulting file to their own proprietary format, usually specific to a particular hardware platform, application, or operating system. This approach generally makes the reuse or repurposing of media extremely difficult. In particular, the compositional metadata (the data that describes the construction of the composition and not the actual program content itself) is not transferable between authoring applications.

The Advanced Authoring Format defines authoring as *the creation of multimedia content including related metadata*. In the authoring process, it is important to record not only the creative decisions that have been made, but also the steps followed to reach the final output, the sources used to create the output, the equipment configuration, intermediate data, and any alternative choices that may be selected during a later stage of the process.

AUID
unique identifier which is a SMPTE Universal Label conforming to SMPTE 298M-1997 or another 128-bit unique identifier

component
basic object that defines essence in a track

composition package
metadata object that describes how to combine and modify content elements and content items to produce a content package

AAF file
storage wrapper data file that stores essence and metadata in objects that conform to the AAF specification

essence
parts of content that directly represent program material, such as audio, video, graphic, still-image, text, or other sensor data

file package
metadata object that describes an essence component stored in a digital form in a file

header
root object of the file that contains the packages and EssenceData objects in the file and defines extensions to the classes used to store objects in the file

interchange object
a set of metadata that includes additional information to assist AAF interchange

key length value (KLV)
a SMPTE metadata binary format

material package
metadata object that specifies association and derivation metadata; it provides a level of indirection between a composition package and a file source package and synchronizes file source packages

metadata
parts of content that contain data used to describe essence or provide information on its use

metadata object (package)
structure that has a globally unique identity and describes essence

object
a unique instance of a data structure defined according to the template provided by its class; each object has its own values for the variables belonging to its class and can respond to the messages (methods) defined by its class

The AAF Model

The Advanced Authoring Format is an industry-driven, cross-platform, multimedia file format, based upon the SMPTE/EBU model, that will allow interchange of data between AAF-compliant applications. There are two kinds of data that can be interchanged using AAF:

1. Audio, video, still image, graphics, text, animation, music, and other forms of multimedia data. In AAF these kinds of data are called **essence** data, because they are the essential data within a multimedia program that can be perceived directly by the audience.
2. Data that provides information on how to combine or modify individual sections of essence data or that provides supplementary information about essence data. In AAF these kinds of data are called **metadata**, which is defined as data about other data. The metadata in an AAF file can provide the information needed to combine and modify the sections of essence data in the AAF file to produce a complete multimedia program.

Together, the essence and metadata describe **content** varying in complexity from simple to highly-structured. Essence, metadata, and content correspond to a hierarchy of classes (see *AAF Class Model and Hierarchy* on page 12).

Essence

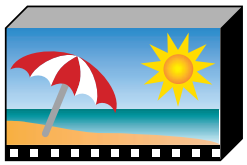
AAF files can describe and contain a broad range of essence types and formats. These essence types include the following:

- Video essence in various formats (RGBA, YCbCr)
- Sampled audio essence in various formats (AIFC, Broadcast WAVE)
- Static image essence
- MIDI music essence
- Text essence in various formats
- Compressed essence formats (M-JPEG, DV, MPEG)

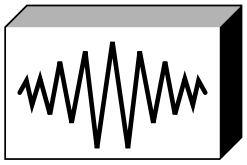
In addition to the essence formats listed above, the AAF standard provides a general mechanism for describing essence formats, and defines a plug-in mechanism that allows applications to be extended to support new types of essence data.

Essence source information describes the format of digital audio and video data, and how the data was derived (the sampling format and compression, if any, used). Source information can also include tape timecode, film edgecode data, and non-standard information such as “smart lens” data.

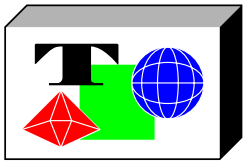
Essence



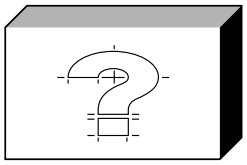
Video Essence
Digital film, HD, NTSC, PAL or other moving picture source (e.g. a source clip from a single video track)



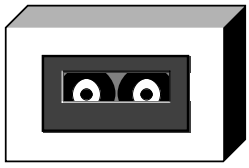
Audio Essence
AIFF, WAV, AU or other digital audio source (e.g. a source clip from a single audio track)



Other Essence
TIF, 3D, JPG, TXT or other digital source (e.g. a titling overlay)



User-Defined Essence
New digital source specified by user (e.g. proprietary format)



Physical Essence
Physical source (e.g. D1 tape)

package
the AAF term for "metadata object"

packageID
value that defines the unique identification of a package

physical source package
package that describes physical media

property
element in a file that has a name, type, and value

relational metadata
metadata that describes how to synchronize or interleave essence

segment
component that has well defined boundaries; a segment can be used without any other components in contrast to a transition, which can only be used in a sequence and need to be surrounded by segments

sequence
an ordered set of components arranged in a sequential order

SDTI-CP (SMPTE 326M)
The latest generation development of the Serial Digital Interface (SDI). SDTI-CP multiplexes both essence and metadata streams together and provides synchronization, all independent of the data type.

source package
metadata object that describes an essence component stored either in a digital form or on a physical media source

static metadata
metadata that describes the edit interchange file as a whole

storage wrapper
persistent storage mechanism for the storage of complex content
NOTE This mechanism allows descriptive information to be stored with the data in such a way that it is possible to query the wrapper file to find out the format of the data and then to use that information to read and interpret the encapsulated data.

track
object in a package that describes essence

Metadata

The number of distinct varieties of metadata is potentially limitless. SMPTE divides metadata into several categories, depending upon its purpose. AAF supports these as properties in the AAF data model.

- **Identification and Location Metadata** – comprises all forms of metadata that can be used to uniquely identify an item, be it essence, object, device or other. It is also used for metadata which is used to locate essence data, metadata or objects.
- **Administration Metadata** – a business metadata class used for the definitions of rights, user access, security classifications, encryption, audience listings and other business information.
- **Interpretive Metadata** – partly for human-orientated metadata types such as names, artists, organisations and classification. It is also used for metadata which defines how to interpret subsequent data types (such as a language descriptor).
- **Parametric Metadata** – describes signal coding parameters of all forms, and device characteristics such as sensor parameters (e.g. focal length) plus device storage and streaming parameters.
- **Process Metadata** – includes all items that describe how essence is assembled, such as editing and compositional metadata. This metadata may be used by a processor to create new material from source material.
- **Relational Metadata** – describes how information is related and provides, in effect, the 'verbs' of a metadata/essence 'sentence'. This metadata is used to describe the relationship between essence types, metadata types and metadata to essence relationships.
- **Spatio-Temporal Metadata** – describes places and time including angles, geo-spatial coordinates, dates, creation times, event times, delays and durations.

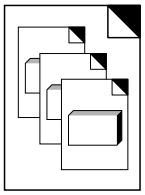
Within each category, metadata may be further divided into subcategories. Some varieties of metadata may be treated as essence in certain circumstances. For example, within an Asset Management system, a sequence of key phrases may be used to index and describe the content, and should be regarded as metadata; but if the same text is converted into captions and inserted into a broadcast television signal, it can be regarded as data essence.

Metadata Characteristics

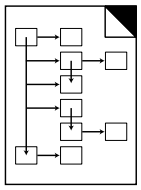
Each of the metadata types described above has one or more characteristics, adding more flexibility to the way in which compositions can be described by AAF.

- **Vital** – metadata that is absolutely necessary for the operation of a system. The specific set of vital metadata may be different for each application, but it always includes at least the essential metadata.

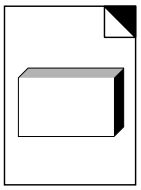
Metadata



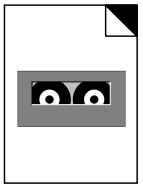
Master Metadata
Information specifying location of File Source metadata packages



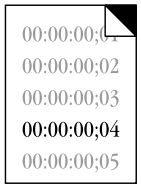
Compositional Metadata
Information describing the structuring and layout of Essence Components



File Source Metadata
Information describing Essence Components
(e.g. essence type)



Physical Source Metadata
Location of video tape or file, plus timecode or edgecode



Relational Metadata
Information related to essence component interaction (e.g. synchronization data)

UMID
a type of Unique Content Identifier described by SMPTE, used to identify all kinds of Interchange Objects in the system, both permanent and transient

variant metadata
metadata that describes an element or subsection within an edit interchange file

vital metadata
metadata required by the system

XML
Extensible Markup Language — a subset of the SGML document language that allows tags to be defined in a Document Type Definition (DTD).

- **Static** – metadata that is related to the whole of a subsection of the Content (for example a Content Item or Content Package)
- **Variant** – metadata that is related to a subsection of the Content (e.g. a single Content Component, a Content Element, or a frame). Variation will frequently be connected to the timing of the Content, but may also be associated with other indexing of the Content.
- **Transient** – some metadata types may intentionally be destroyed after they have served their useful purpose. Examples include QoS Control, Machine Control, Error Management and Encoder Control
- **Permanent** – examples include essential metadata such as Unique Material Identifiers (UMIDs)

Metadata may be kept with the associated essence or kept elsewhere. Factors contributing to the choice include allocation of available bandwidth, ease of access, and concern for systems reliability and error recovery.

These characteristics of metadata are recorded in the relevant standards documents and in the SMPTE Registry. Other characteristics may well be identified in the future.

Metadata Sets

Metadata may be grouped into sets that are referenced as a single item. Such groupings are known as **metadata sets** (see *interchange objects* on page 7). Sets can be referenced by a single key value rather than by identification through each individual metadata item; individual items within the sets may be identified with sub-keys, or implicitly within fixed formatting of the set.

SMPTE defines the syntax of collections of metadata items called Sets. In AAF, these are used extensively to represent Interchange Objects. Packages are specific kinds of Interchange Objects.

A Note About AAF Nomenclature

In the source code, Packages and Tracks are called "Metadata Objects (MOBs)" and "Slots". Early revisions of the Specification and API Reference use these more software-oriented names exclusively; later revisions include the more generic terms. Other terms that have been updated include:

Updated Term	Software-Oriented Equivalent Term
Package	Mob (MetadataObject *or* Object)
Material Package	Master Mob
File Package	File Source Mob
Source Package	Physical Source Mob
Composition Package	Composition Mob
Track	Slot

" The innovative AAF specification supports user-defined extensions to the basic format, and the delivery of files that include not only these new extensions, but also their full descriptions. Interaction between AAF-compliant application software and a www-based registry server allows systems to define and publicize templates for files that are tailored for specific end-use applications, such as transmission, enhanced interactive television, DVD creation and streaming media authoring, and then to produce and validate such files."

Brad Gilmer,
AAF Association

" SMPTE has been trying to find a way to speed up the standardization process. The thing about standards is that it takes so long to create them that, by the time they're done, technology has often marched on. Now we're looking at a 90- to 180-day timeframe to be able to extend standards, and actually to make them available electronically on the internet."

Merrill Weiss, SMPTE

Content

In the simplest AAF structure, metadata and essence combine to form a **content element**. Content elements can be grouped (with metadata describing the grouping) into **content items**. Content items and content elements can then be arranged into **content packages**—the AAF equivalent of an authored multimedia composition. Any of these can be represented as a binary file by the addition of a **wrapper**.

The construction of the wrappers requires additional items of data. This data is referred to as **overhead**. Overhead includes such things as flags, headers, separators, byte counts, and checksums.

These content structures cover the basic needs of describing and containing source material, finished material, sync relationships, cut lists, and play lists. There are, however, many other kinds of elements needed to build programs, including effects descriptions, compositing information, and entire catalogs of source material.

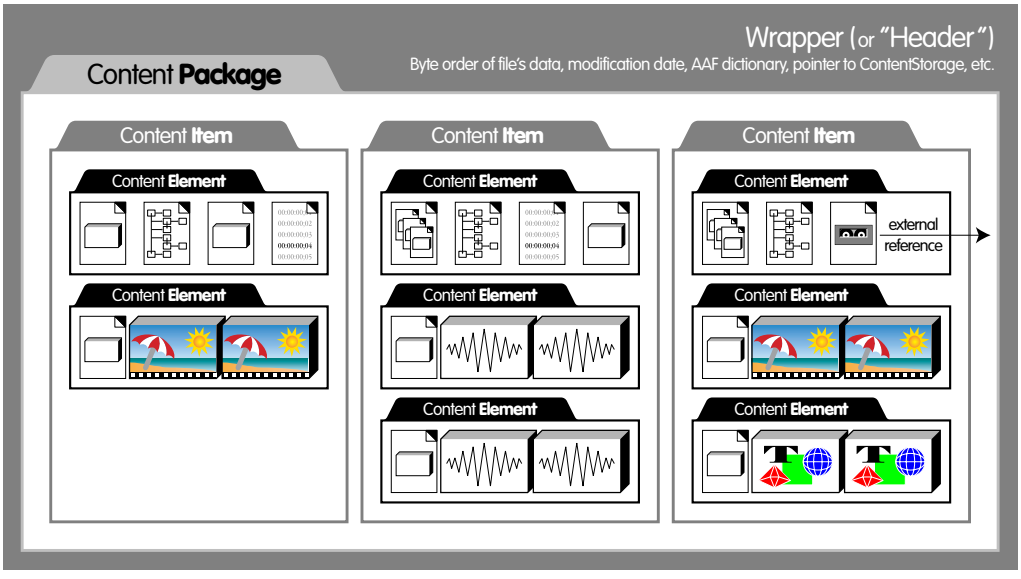
The various kinds of elements are catalogued and described by several dictionaries maintained by SMPTE (e.g. the **metadata dictionary**, which contains all the basic descriptive elements).

One of the SMPTE dictionaries describes sets of metadata—items grouped together for particular application purposes. "Sets" is the generic term. Specific sets are called **interchange objects**.

In the SMPTE-derived AAF data model, all the constituent components of content are represented as various classes of interchange object. The data model describes how interchange objects may be constructed, and how they can contain or reference (point to) other

" …devices will be able automatically to download extensions to standards. When they encounter some metadata that they don't recognize, they can automatically go find the definition, download it and work with it right away by virtue of this much speeded-up standardization extension process."

Merrill Weiss, SMPTE



Typical SMPTE/EBU File
The SMPTE/EBU definition of Content Elements, Items, Packages and Wrappers is the basis for AAF. A rough correspondence to familiar terms is that a Content Element is a source clip from a single video or audio track, a Content Item is a synchronized master set of clips encompassing several tracks, and a Content Package is an edited composition of several clips.

interchange objects. The resulting data structures, known as **complex content packages**, can describe arbitrary levels of editing and processing in the creation of a finished program.

Unique Content Identifiers

Unique Content Identifiers identify a specific piece of content independent of where the content is stored and independent of whether it is a copy or the original material. They are used to link between uses of content and the content itself, through the medium of an asset-management database.

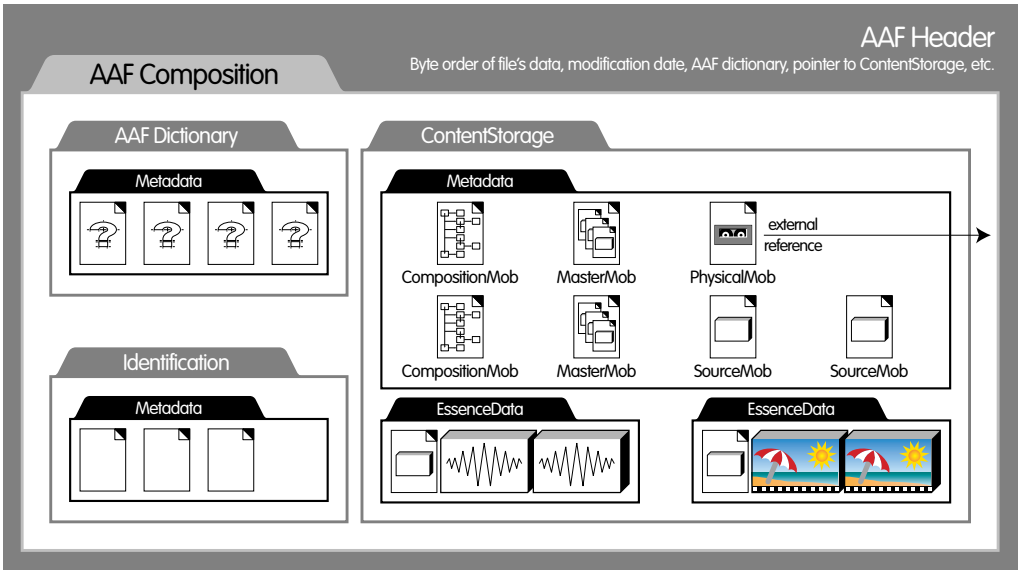
AUIDs are a type of unique identifier used by AAF. They have a defined mapping with SMPTE labels, and can be generated either by a client application or from within the AAF SDK.

A **UMID** is a Universal Material Identifier that SMPTE uses to identify essence and metadata. A UMID contains an AUID as one of its fields.

" Since the announcement of the AAF initiative, the standardization work in SMPTE has proceeded to form a consensus about the formatting and the basic dictionary of global metadata items. The AAF initiative provides a tangible implementation of these ideas"

Oliver Morgan, Avid

Typical AAF File
In the EBU/SMPTE data model used by AAF, all the constituent components of content are represented as various classes of interchange object. The data model describes how interchange objects may be constructed, and how they can contain or reference (point to) other interchange objects. The resulting data structures can describe arbitrary levels of editing and processing in the creation of a finished program, and are known as **complex content packages**.



Interchanging Compositions via AAF

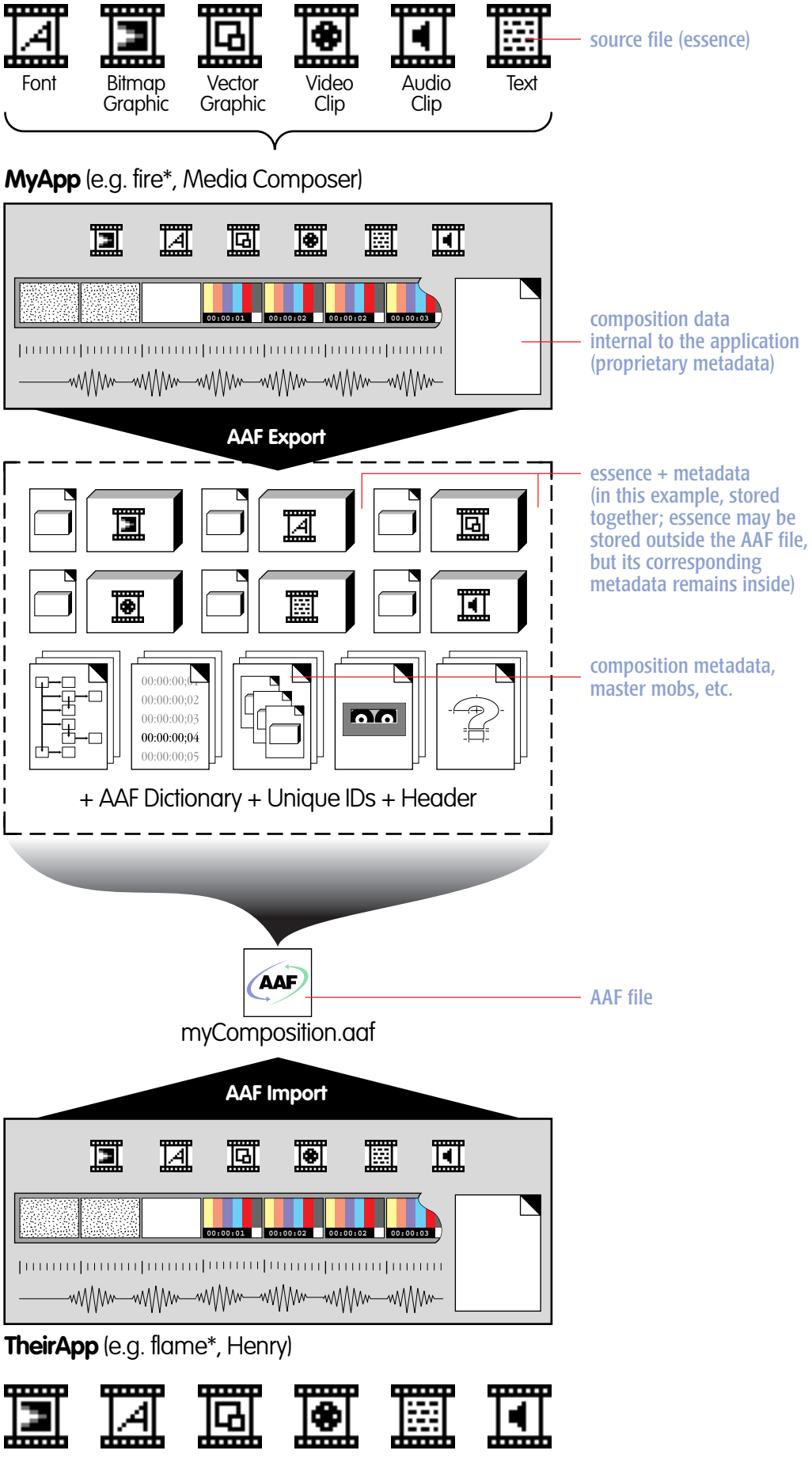
The AAF format allows end users to bring the many disparate sources of a composition together (if they choose) into one easily managed file. In this example, a series of source files are assembled into a composition using *MyApp* software. AAF-compliant MyApp can export the composition so that it will be readable by *TheirApp* , or any other AAF-compliant software. Any "special knowledge" that forms part of the original composition is saved in such a way that, if *TheirApp* cannot understand it, the "special knowledge" can be safely ignored while remaining integral to the file. No information is lost!

Upon export, each source (or "essence") file is wrapped as an AAF object. The wrapping process includes the creation of metadata that describes the essence file. In the AAF model, each wrapped essence and metadata pair is referred to as a "content element".

Content elements are organized into a structure called a "content item". The content item includes more metadata that describes the constituent content items, as well as their relationship to each other.

Finally, a file is created[1] that ties one or more content items into a "content package". The wrapper for this file identifies it to compliant applications as AAF. By first reading the master metadata in "myComposition.aaf", *TheirApp* is able to break it down into its constituent content items and elements. The metadata associated with each of these enables the composition to be mapped to *TheirApp's* native architecture (except for "special knowledge", which may be ignored). AAF provides a way to avoid having to target the lowest-common denominator while allowing simpler applications to use the information they understand.

[1] In practice, more than one .aaf file may be created, due to the size of typical high-end compositions.



The AAF File Format

Microsoft Structured Storage (MSS), one of the technical underpinnings of AAF, refers to a data storage architecture that uses a "file system within a file" architecture. This container format is a public domain format, allowing interested parties to add future developments or enhancements in a due process environment. Microsoft is specifically upgrading the core technology compound file format on all platforms (Microsoft Windows®, Apple® Macintosh®, UNIX) to address the needs of AAF.

AAF files are able to store a variety of media file formats and the complex metadata that describes their usage. The file format provides a structured container for essence and metadata in a single object-oriented model for interchange. The AAF format encapsulates essence—preserving its file-specific, intrinsic information—as well as the authoring information (in and out points, volume, pan, etc.) describing the essence and any interactions with it. It supports efficient playback and incremental updates. AAF is scalable, making it equally suitable for very high-end professional applications as well as those at the consumer level.

Other important features include:

- The ability to retain information about original sources for an edited program—this makes it possible to easily regenerate a program from its original sources, or to re-purpose sections of a program
- Modification history, making it possible to track the applications that have been used to create and modify the program
- References to external media files, with files located on remote computers in heterogeneous networks
- An extensible video and audio effects architecture with a rich set of built-in base effects
- Support for a cross-platform binary plug-in model
- Support for output in XML format

" The AAF should provide an end to the islands of incompatible automation that plague our industry, and the beginning of complete digital media environments built from products from multiple vendors."

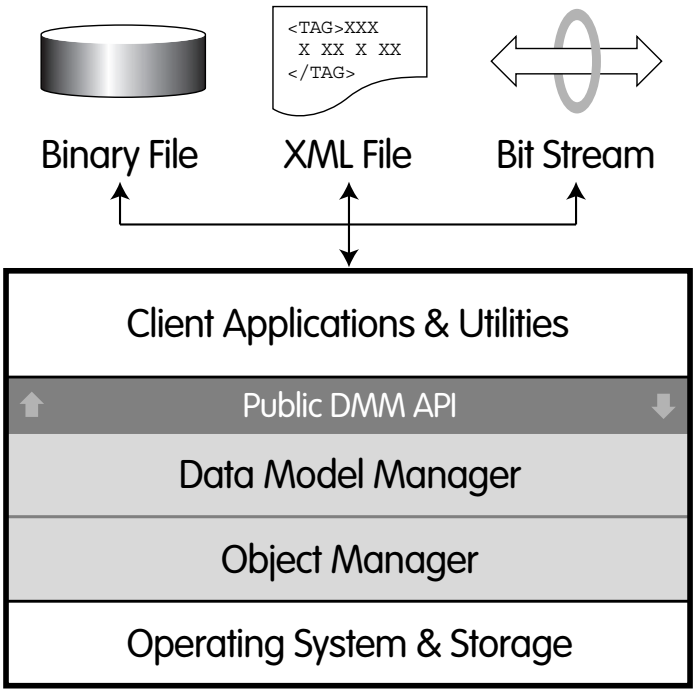
David Dale, Avid

Content Delivery

AAF is an authoring format, and does not specifically address the issues associated with delivery. Nonetheless, the content created using AAF in the authoring process will be delivered by many different vehicles, including broadcast television, packaged media, film, as well as over private networks and the internet. These delivery vehicles will use data formats such as baseband video, MPEG-2 Transport Stream, and the Advanced Streaming Format (ASF). These formats do not need as rich a set of metadata as that used during the authoring process. AAF files can be optimized for delivery by "pruning", or stripping out this metadata.

Software Architecture

The most important goal of the AAF development was to make the SMPTE Data Model readily accessible to software developers throughout the industry and usable by a broad range of authoring applications. The best vehicle for this is a software developers' kit (SDK) to present the SMPTE Data Model to developers through an API supported on many computer platforms. Two of the most important requirements are for cross-platform implementation and for user extensibility.



Client Applications & Utilities

Client applications implement all the macroscopic behaviours of the system, using the fundamental services provided by the Data Model Manager, and other system services.

Utilities are specific client applications with limited but important functionality (e.g. File Dumpers, Validators, Generators). Typically they will be used for system administration, testing, or to help client developers.

Another kind of client application converts between the SMPTE Data Model and other formats (e.g. serial interconnect formats, proprietary formats, XML, and SDTI-CP).

Conversion functions are created as pluggable component software, which can be made accessible to multiple clients.

Data Model Manager

The Data Model Manager (DMM) implements Persistence, Transaction, and Navigation services upon Interchange Objects, which it exposes to clients through the API.

Object Manager

The Object Manager (OM) provides the basic functions of Saving and Restoring objects and sub-objects and maintaining the relationships between them.

Storage System

The Storage System underlying the Object Manager is normally one of the file systems provided by the OS. In the AAF SDK, this function is provided by Microsoft Structured Storage.

Operating System Platform

Underlying all the other subsystems is the Operating System. One of the challenges in designing the DMM and OM was to keep them separable from the Operating System, in order to serve the cross-platform interoperability requirements of the clients.

Public Application Program Interface (API)

The public API is the aspect of the Data Model Manager that all client applications see, and that treats all potential clients equally.

- It is written in Interface Definition Language (IDL), to permit bindings to different languages (e.g. C, C++) and object brokers (e.g. Component Object Model).
- It provides basic services: persistence (save and restore), transaction (add, modify, delete), accessors (get, set), and navigation (traversal, iteration, query).
- It makes full use of polymorphism.
- It has a very regular, predictable structure, to encourage consistent coding style and allow extension over time.
- It provides clear mechanisms for extension of the Data Model, so that new object types can be linked into the API without causing revision or recompilation of the kernel software.

Data Model Manager (internal)

Beneath the public API there are various interfaces and implementation helper functions that are not expected to be directly called by the client. It is here that much of the design value of the Data Manager is concentrated.

One of the benefits of using IDL to define the public API is that the unpublished implementation details are defined separately, reducing the temptation for clients to use an internal function and risk less than full error checking.

Object Management (internal)

The architecture separates Data Model Management from generic Object Management. The interface between these two subsystems is not public; the DMM interface exposes the OM interface polymorphically through the DMM API.

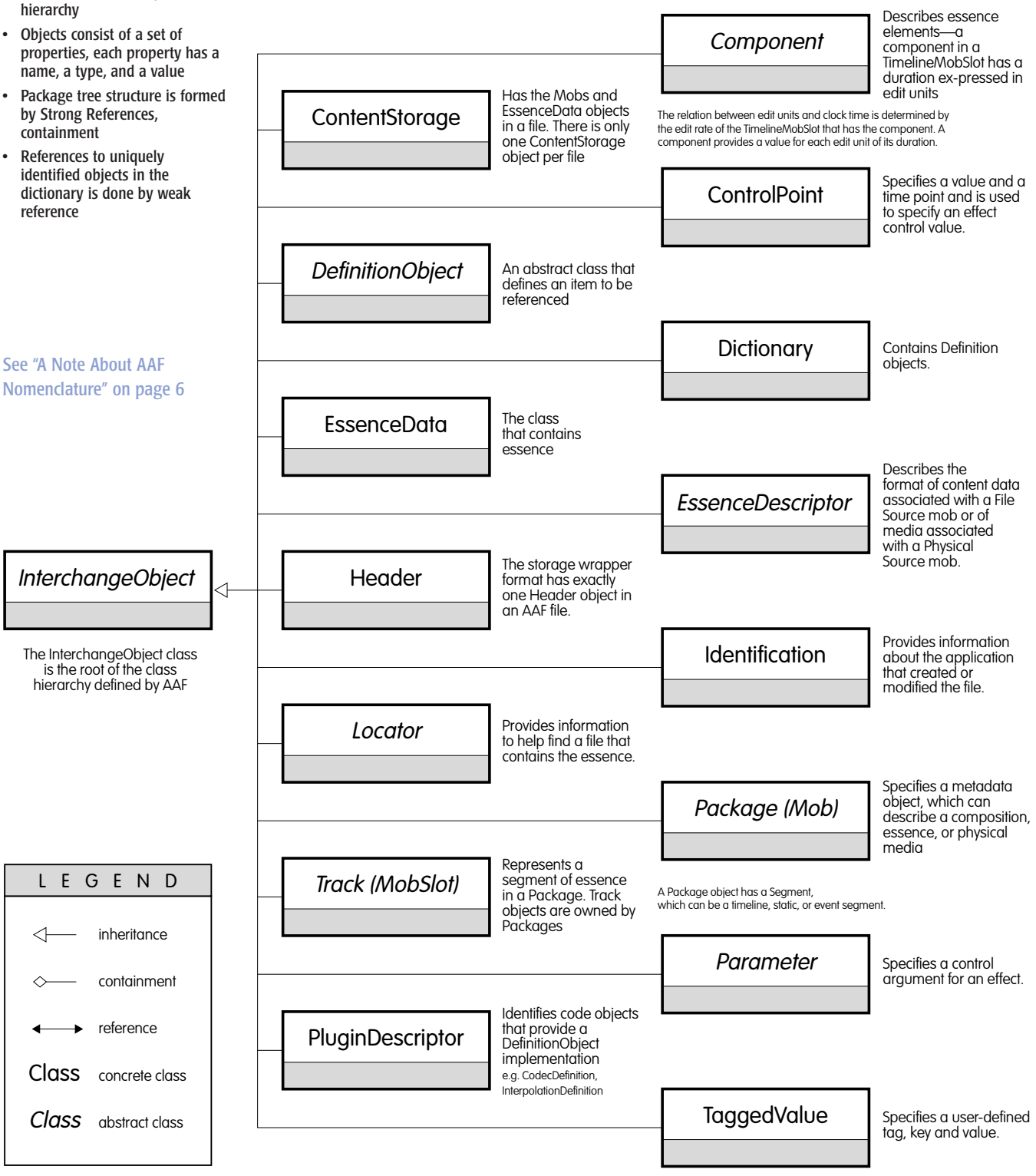
AAF Class Hierarchy

AAF Object Model Structure

- Dictionary contains definitions, such as Class Definition
- Packages are composed of a tree structure of object instances. Each object instance belongs to a Class, which is defined by a Class Definition
- Classes are defined by a class hierarchy
- Objects consist of a set of properties, each property has a name, a type, and a value
- Package tree structure is formed by Strong References, containment
- References to uniquely identified objects in the dictionary is done by weak reference

The AAF classes are used to describe multimedia compositions and data. A class specifies an AAF object by defining what kind of information it may contain and how it is to be used. Each AAF class inherits from one immediate superclass, thereby avoiding the problems associated with multiple inheritance.

See "A Note About AAF Nomenclature" on page 6



" The fact that the AAF is actively supported by manufacturers, broadcasters, and post-production houses shows a common interest. Everyone involved stands to benefit from co-operating on a pragmatic standard."

Mark Horton, Quantel

AAF Object Model Basics

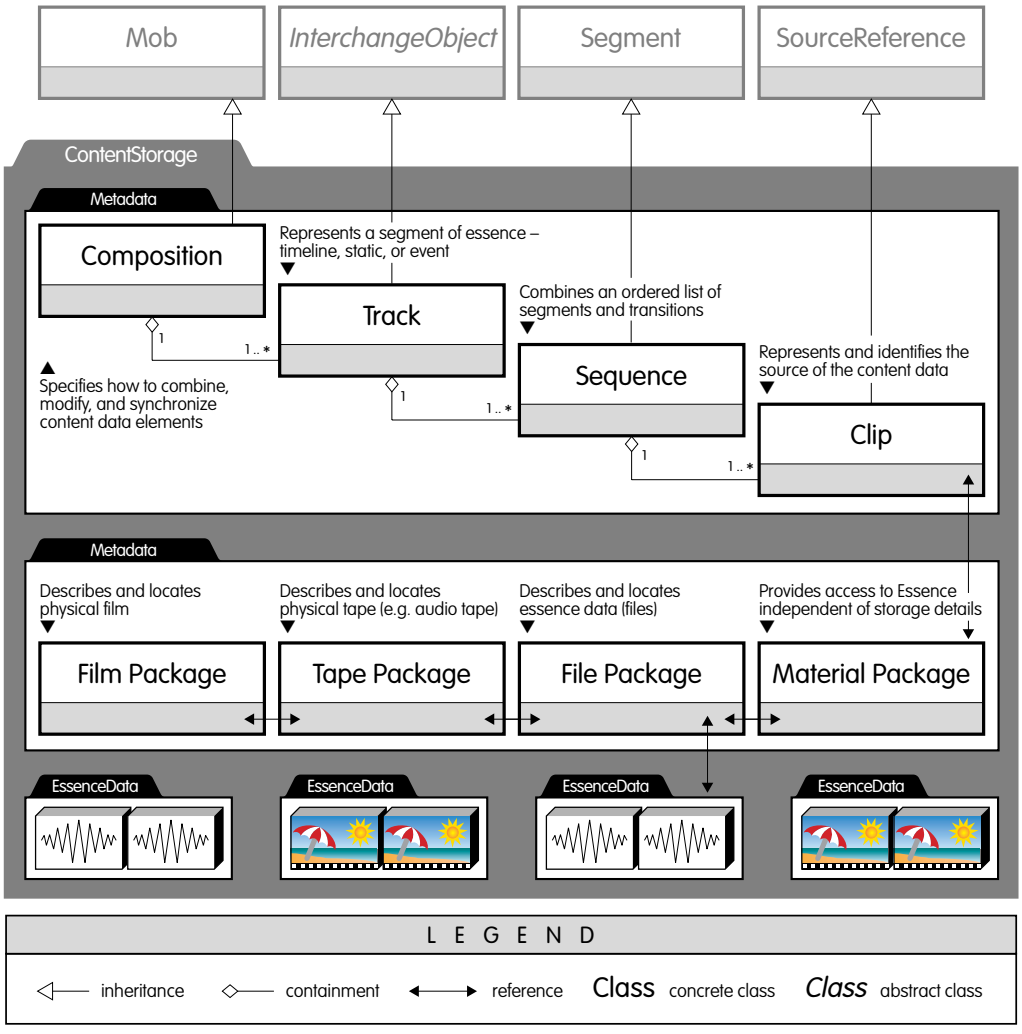
Composition Packages describe editing/creative decisions containing tracks with sequences and effects. **Material Packages** provide a level of indirection that allows users to work with virtual playable audio and video, while hiding storage details (such as that video and audio are in separate files, that there may be multiple versions of the video with different compressions, etc.)

File Packages describe the digital playable audio and video stored in files. **Tape** and **Film Packages** describe the original physical media. Packages contain **T**racks.

A **Sequence** contains Source Clips, Effects, and Transitions that are played in sequential order. The start time of each item in the sequence is determined by when the previous item ends (unlike an EDL where each clip has a specified start time).

The derivation/transformation of the media is described by the chain of Source Clip to Package. A SourceClip in a Composition Package points to the previous generation Package (Master, File, Tape, Film) that describe the source of the media.

See "A Note About AAF Nomenclature" on page 6



" The source code of AAF reference implementation is fully available, free of charge and royalty, to anybody in the entire universe."

Oliver Morgan, Avid

The AAF Software Development Kit

The AAF SDK Reference Implementation is an object-oriented, platform-independent programming toolkit (with supporting documentation) that allows client applications to access the data stored in an AAF file. There are several reference platforms on which the SDK has been built and tested. The toolkit is available in source form.

The SDK also enables developers to create alternative implementations that access the data in an AAF file, based on the information in the AAF Object Specification and the AAF Low-Level Container Specification.

Microsoft's Component Object Model (COM) is employed by the SDK as a programming interface for client applications. COM provides mechanisms for components to interact independently of how the components are implemented.

AAF defines a base set of built-in classes. These built-in classes can be used to interchange a broad range of data between applications. Where applications have additional forms of data that cannot be described by the basic set of built-in classes, AAF provides a mechanism to define new classes.

Open Source

For any new software standard to be acceptable to a wide audience and portable to new platforms, the source code must be available without barriers to developers. For the AAF SDK, this has been achieved. Patent and intellectual property issues have been resolved, and the source code is available for download and compilation, without license fee or royalty. The AAF Association aims to establish all the AAF SDK as Open Source with collaborative development of new versions. The Open Source process will get underway with the third developer release (DR3) in Summer 2000.

" The AAF is a major part of the vision that Microsoft has been investing heavily in for years. Having a standard interchange format brings tremendous benefits to everyone in our industry."

Tom MacMahon, Microsoft

AAF Documents

The Advanced Authoring Format documentation includes:

- **The AAF Object Specification**
- **The AAF SDK Reference Implementation Methods and Interfaces**
- **The AAF SDK Reference Implementation Developers' Guide**
- **The AAF Low-Level Container Specification**

Installation Configurations

1. **End User**: This kit is intended for end users who are using applications that support AAF. This kit includes the libraries needed to run applications that use AAF. It also includes an AAF/OMF converter.
2. **Application Developer**: This kit is intended for developers who are creating applications that use the AAF SDK. This kit installs the header files and libraries needed to create application programs. It also includes source code of example and test programs and binaries of the example, test, and utility programs. Note that application developers working with prerelease kits may choose to use the Platform Developer Kit to make it easier to debug code.
3. **Platform Developer**: This kit is intended for developers who are porting the AAF Reference Implementation to a new platform or who are creating plugin codecs. This kit installs the complete AAF SDK source tree with projects, binaries, and sources. It includes sources for the examples, tests, and utilities as well as the reference implementation of the SDK.

Files Provided in the SDK Platform Developer Release

- Release notes, including license agreements, build instructions, and a list of known bugs
- Header Files and Pre-built DLLs, Libs, and EXEs/Apps
- CodeWarrior® projects for AAF Reference Implementation on Macintosh PPC (included only with Macintosh® installer)
- MS Visual Studio workspace and projects for AAF Reference Implementation on Windows NT (included only with Windows NT® installer)
- Source files for low level dumper utility program
- AAF Specification, AAFGuide and API documentation
- C++ source files for the examples (client applications)
- Source tree (code) for AAF Reference Implementation
- C++ source code that tests that the reference implementation is executing correctly
- Source files for the utility programs

SDK System Requirements

Windows

Windows NT v4.0 with Service Pack 4 or Windows® 2000 Professional Prerelease (build 2000)
Microsoft® Visual C++® 6.0

Macintosh

Macintosh PPC with Mac® OS 7.6 or higher
CodeWarrior Pro3 updated to CodeWarrior IDE 3.1
Required Libraries and Headers

UNIX/IRIX/Linux

SDK available with DR3 (July 2000)

" The BBC is committed to improving the links between different stages of the programme-making process. With solid industrial backing and the AAF Association to guide its development, the AAF format is on course to deliver the dream of seamless transfer of content between systems."

Phil Tudor, BBC

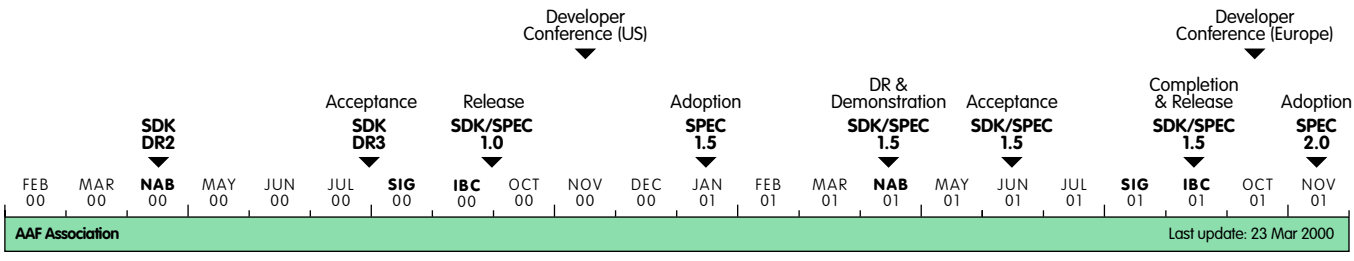
About the AAF Association

Founded in March 1998 and incorporated in February 2000, the AAF Association Inc. is a broadly-based trade association intended to promote the development and adoption of AAF technology throughout the media industry. With representatives from some major players in the industry (Avid, British Broadcasting Corporation, CNN/Turner, Discreet, Matrox, Microsoft, Pinnacle Systems, Quantel, Sony, U.S. National Imaging & Mapping Agency, and 4MC), the AAF Association intends to help deliver the full benefits of digital media to content creators including film, television, and post-production professionals.

Membership in the AAF Association is open to any interested parties. For more information, visit **www.aafassociation.org**.

Development Timeline

Initial work on what was to become the AAF software development kit predates the EBU/SMPTE TFHS final report. Developer Release 1 (DR1) was issued in October 1999. DR2 of the SDK Version 1.0 will be available shortly. A third Developer Release (featuring support for UNIX, XML, and various codecs) will mark the start of final testing in products in mid-summer 2000, and the End User Version 1 will be released in early Autumn 2000.



" Working together, we can help the industry to realize the full benefits of going digital. For the broadcast and post-production community, the AAF Association provides a unique opportunity to influence the working practices of the future. For manufacturers, the AAF Association means participating in a project that will have a fundamental impact on the future markets for their products."

Brad Gilmer,
AAF Association

Top 10 Reasons for Joining the AAF Association

1. AAF provides a common format for interchanging digital content and metadata that removes barriers to workflow, improves the creative process, and saves countless hours of labour.
2. AAF allows for smooth interchange of content and metadata across platforms, between applications, and between vendors.
3. The AAF SDK toolkit is available, without licensing fees, to software developers who can leverage this technology. Ultimately, the AAF SDK will be established as open source.
4. AAF addresses the issue of archiving of program content, which is a critical application in fast-changing internet/media area. For example, the United States National Science Foundation says, "Digital Libraries have been identified as a "National Challenge"… National Challenges are fundamental applications that have broad and direct impact on the United States' competitiveness and the well-being of its citizens."[1]

   [1] **"NSF announces awards for Digital Libraries Initiative"**, http://walrus.stanford.edu/diglib/pub/nsf.announce.html
5. Users can expect long term personal efficiency gains by the introduction of digital video and related metadata both in the workplace and in homes. Such gains will be substantially increased if links between the video and other program elements can be maintained and described.
6. AAF is the most comprehensive format available, providing its adoptees with the potential to penetrate the widest market possible.
7. AAF, like Quicktime, allows stored data to contain information about sequencing, timing, and special effects, but is much more geared to formal compositions.
8. The AAF SDK has an automated test suite
9. AAF is extensible—as new codecs, transitions, effects and plug-ins come into being, AAF will adapt and grow with the times.
10. AAF is a nonproprietary format created by broad industry consensus, but is independent of any one player.

For More Information

www.aafassociation.org
The AAF Association official site

www.ebu.ch/pmc_tfbrief.htm
The Joint EBU/SMPTE Task Force Technical Briefing held at IBC'98

www.ebu.ch/pmc_es_tf.html
The Joint EBU/SMPTE Task Force Final Report

www.eet.com/news/97/966news/global.html
Bringing "order to the chaos" of the digital-TV transition (J. Yoshida)

www.techweb.com/se/directlink.cgi?EET19970630S0001
RFTs focus on digital-TV interoperability (June 30, 1997)

www.digitaltelevision.com/dtvbook/appendixc.shtml
Metadata & Content: A Guide for Video Pros (L. CasaBianca & C. M. Okon)

www.avid.com/news/press_releases/product_news/aaf.html
Avid OMFI Selected as Foundation For Advanced Authoring Format

www.dv.com/magazine/1998/0798/0798author.html
The Quest for Full Media Interchange: An Overview of AAF

http://www.dv.com/magazine/1999/0899/roundtables0899.pdf
Metadata for the Masses — exchanging media and project information

Membership in the AAF Association is open at various levels to any interested parties.

To learn more, please contact us at:

info@aafassociation.org