SMPTE ST 2067-3:2016
Revision of
SMPTE ST 2067-3:2013

SMPTE STANDARD

Interoperable Master Format — Composition Playlist

Table of Contents

Page

Approved
May 18, 2016

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 Part XIII of its Operations Manual.

SMPTE ST 2067-3 was prepared by Technology Committee 35PM.

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 IMF is an interoperable file-based framework designed to facilitate the management and processing of multiple content versions (airline edits, special edition, languages…) of the same high-quality finished work (feature, episode, trailer, advertisement, etc) destined for distribution channels worldwide. Each content version is embodied in a composition, which combines metadata and essence.

The Composition Playlist defines the playback timeline for the Composition and includes metadata applicable to the Composition as a whole. It is a human-readable structure expressed using XML and specified using XML Schema. It includes multiple extension points supporting both backward- and forward-compatibility.

The Composition Playlist is not designed to contain essence but rather reference external Track Files that contain the actual essence. This allows multiple compositions to be managed and processed without duplicating the essence in common. For convenience, metadata contained in the Track Files is exposed in the Composition Playlist (see Section 6.1.10.)

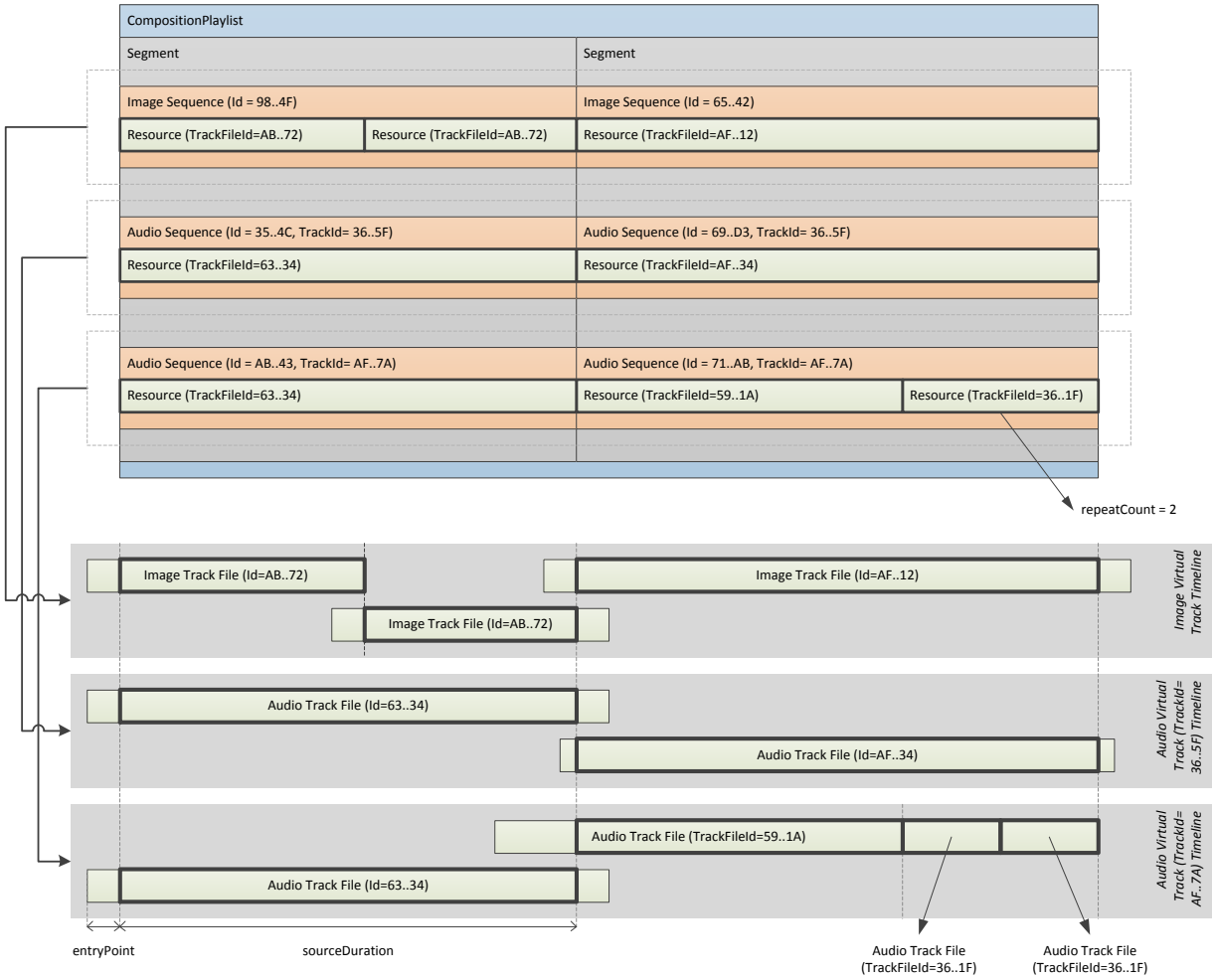

Figure 1 – Sample Monoscopic Composition Playlist
Only the first and last bytes of UUIDs are represented.

The Composition timeline[1] is illustrated in Figure 1. It consists of a list of Segments that are reproduced sequentially and without gaps. The number of Segments and the duration of each Segment are determined by the content provider to fit both the particular workflow and content type. For example, a Segment can be the same duration as a reel or it can be the running time of a television program between commercials. Each Segment consists of a list of Sequences that are played in parallel and have the same duration. Each Sequence is associated with a single kind of essence or metadata and consists of a list of Resources that are reproduced sequentially.

Each Resource references a playable region of essence (audio, image, data, etc.) or metadata contained in an underlying Asset. The Composition timeline makes no assumption on the format of the asset, other than its ability to reproduce essence or metadata in the playable region.

[1] The Composition Playlist does not use timecode to synchronize Assets but defines a canonical synthetic timecode, which implementations can use.

Assets will typically be external Track Files, which can be referenced by multiple Resources, within and outside a particular Track. In Figure 1, Audio Track File Id=63..34 is referenced by Audio Sequences Id= 35..4C and Id=AB..43. Image Track File Id=AB..72 is referenced twice within the first Image Sequence. A Track File can be repeated multiple times within a single Resource. This allows, for instance, gaps in Audio Tracks to be filled by repeating a Track File containing a few samples of silence. If an Asset contains a very small amount of XML information and is referenced by a single Resource, it can be included directly within the Composition Playlist structure, e.g. see MarkerResourceType in Section 6.13.

The Composition Playlist does not define any form of transition or overlap between resources. In other words, the essence contained in Assets is prepared for artifact-free reproduction across these boundaries.

Temporal offsets and durations on the Composition timeline are expressed as integer multiples of Edit Units, the smallest temporal increment accessible on the timeline and the inverse of the Edit Rate. Edit Rates and Edit Units can be defined independently for the Composition and each Resource. The Composition Edit Rate will generally correspond to the image frame rate.

To support a range of applications within its stated scope, the Composition Playlist allows flexibility in the nature and structure of Sequences, Resources and Track Files. Applications are therefore expected to constrain these characteristics to fit their respective requirements. Similarly, the method by which essence is wrapped in an Asset is left to the specification defining the Asset.

1 Scope

This standard specifies the Composition Playlist (CPL) for the Interoperable Master Format (IMF). The Composition Playlist is a representation of a single version of a finished IMF composition (feature, episode, trailer, advertisement, etc). It contains the information necessary to describe the composition and synchronize its underlying essence; e.g., for playout or transcoding. It is an extensible, human-readable structure designed for file-based operations.

2 Conformance Notation

Normative text is text that describes elements of the design that are indispensable or contains the conformance language keywords: "shall", "should", or "may". Informative text is text that is potentially helpful to the user, but not indispensable, and 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 will never 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 which, 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 recommended practice are encouraged to investigate the possibility of applying the most recent edition of the standards indicated below.

[1)] [SMPTE ST 12-1:2008] SMPTE ST 12-1:2008, Television — Time and Control Code

[SMPTE ST 433] SMPTE ST 433:2008 D-Cinema — XML Data Types

[RFC 2045] Internet Engineering Task Force (IETF) (November 1996). RFC 2045 — Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies

[RFC 2046] Internet Engineering Task Force (IETF) RFC2046 (November 1996) Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types

[RFC 2141] Internet Engineering Task Force (IETF) (May 1997) RFC 2141 — URN Syntax

[RFC 2396] Internet Engineering Task Force (IETF) (November 1996). RFC 2396 — Uniform Resource Identifiers (URI): Generic Syntax

[RFC 4634] Internet Engineering Task Force (IETF) (July 2006). RFC 4634 — US Secure Hash Algorithms (SHA and HMAC-SHA)

[RFC 5646] Internet Engineering Task Force (IETF) (September 2009). RFC 5646 — Tags for Identifying Languages

[XML] World Wide Web Consortium (W3C) (4 February 2004). Extensible Markup Language (XML) 1.0 (Third Edition)

[XML Namespaces] World Wide Web Consortium (W3C) (8 December 2009). Namespaces in XML 1.0 (Third Edition)

[XML Schema Part 1: Structures] World Wide Web Consortium (W3C) (28 October 2004). XML Schema Part 1: Structures (Second Edition)

[XML Schema Part 2: Datatypes] World Wide Web Consortium (W3C) (28 October 2004). XML Schema Part 2: Datatypes (Second Edition)

[XML Digital Signature] World Wide Web Consortium (W3C) (12 February 2002). XML-Signature Syntax and Processing

[1)] **Note:** The reference to the superseded version of this standard is intentional. The current version was not reviewed by the Technical Committee to verify if it is appropriate for use with this document.

4 Glossary

4.1 Asset: Metadata or essence underlying a Resource. A Track File is one kind of Asset.

4.2 Composition: Embodies a version of a complete work, combining metadata and essence.

4.3 Edit Rate: A number of Edit Units to be reproduced per second.

4.4 Edit Unit: The smallest temporal increment of access to a timeline.

4.5 Essence: Content, e.g. image, audio or data essence, meant for presentation.

4.6 IMF: Acronym for Interoperable Master Format.

4.7 Main Titles: A credit sequence generally shown near the beginning of a motion picture.

4.8 Metadata: Data about data or data describing other data. Information that is considered ancillary to or otherwise directly complementary to essence. Information that is useful or of value when associated with the essence being provided.

4.9 MXF: Material Exchange Format.

4.10 Native Duration: The total number of Resource Edit Units in an Asset.

4.11 Native Start Point: The first Edit Unit of an Asset.

4.12 Playable Region: A contiguous sequence of Resource Edit Units within an Asset that is intended to be reproduced. An Asset can contain Edit Units before and/or after the Playable Region.

4.13 Resource: A portion of an Asset selected for reproduction on the Composition timeline.

4.14 Sequence: An ordered collection of Resources to be reproduced sequentially.

4.15 Segment: A collection of Sequences intended to be reproduced in parallel.

4.16 Track File: An Asset consisting of a file containing essence of a single kind, e.g. as audio, image or data essence.

4.17 UUID: Acronym for Universal Unique Identifier.

4.18 XML: Acronym for extensible Markup Language.

5 Instance

A Composition Playlist instance shall be an XML document, as specified in [XML], that consists of a single CompositionPlaylist element (see Section 6.1).

5.1 Schema

Each Composition Playlist instance shall conform to the XML schema definitions (see [XML Schema]) found in this specification. In the event of a conflict between schema definitions and the prose, the prose shall take precedence.

Table 1 specifies the XML schema root element.

Table 1 – XML Schema root element definition

```
<xs:schema targetNamespace="http://www.smpte-ra.org/schemas/2067-3/2016"
                   xmlns:cpl="http://www.smpte-ra.org/schemas/2067-3/2016"
                   xmlns:dcml="http://www.smpte-ra.org/schemas/433/2008/dcmlTypes/"
                   xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
                   xmlns:xs="http://www.w3.org/2001/XMLSchema"
                   elementFormDefault="qualified" attributeFormDefault="unqualified">
  <xs:import namespace="http://www.smpte-ra.org/schemas/433/2008/dcmlTypes/" />
  <xs:import namespace="http://www.w3.org/2000/09/xmldsig#" />
  <xs:element name="CompositionPlaylist" type="cpl:CompositionPlaylistType"/>
<!-- schema definitions found in this document excluding this one -->
</xs:schema>
```

The namespace prefixes used in XML Schema definitions herein are not normative values and implementations shall perform correctly with any XML compliant prefix values. The XML Schema definitions found in this specification include elements specified in [XML Digital Signature] and [SMPTE ST 433].

A schema instance that consolidates all inline schema definitions made in this document is provided for convenience in Annex B. In case of conflict, the schema definitions included herein shall take precedence.

5.2 Character Encoding

Composition Playlist instances shall be encoded using the UTF-8 character encoding.

5.3 MIME Type

The MIME type, as defined in IETF RFC 2046, of a Composition Playlist instance shall be text/xml.

5.4 Structure Versioning

The target namespace specified in Section 5.1, i.e. the value of the targetNamespace attribute of the schema element, shall only be used by Composition Playlist instances that conform to this specification as expressed by the combination of its prose and schema definitions.

Instances using specifications that modify the schema definitions or the semantics of the elements defined herein, including future versions of this specification, shall use a different namespace and no two distinct schemas shall have the same target namespace. As such, the target namespace allows implementations to unambiguously identify the defining specification of a Composition Playlist instance.

6 Structure

In order to avoid duplication between text and schema, the cardinality and default values of elements are specified in the schema only.

6.1 CompositionPlaylistType

Table 2 – CompositionPlaylistType schema definition

```
<xs:complexType name="CompositionPlaylistType">
  <xs:sequence>
    <xs:element name="Id" type="dcml:UUIDType" />
    <xs:element name="Annotation" type="dcml:UserTextType" minOccurs="0" />
    <xs:element name="IssueDate" type="xs:dateTime" />
    <xs:element name="Issuer" type="dcml:UserTextType" minOccurs="0" />
    <xs:element name="Creator" type="dcml:UserTextType" minOccurs="0" />
    <xs:element name="ContentOriginator" type="dcml:UserTextType" minOccurs="0" />
    <xs:element name="ContentTitle" type="dcml:UserTextType" />
    <xs:element name="ContentKind" type="cpl:ContentKindType" minOccurs="0" />
    <xs:element name="ContentVersionList" minOccurs="0">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="ContentVersion" type="cpl:ContentVersionType"
                    maxOccurs="unbounded" />
        </xs:sequence>
      </xs:complexType>
    </xs:element>
    <xs:element name="EssenceDescriptorList" minOccurs="0">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="EssenceDescriptor" type="cpl:EssenceDescriptorBaseType"
                    maxOccurs="unbounded" />
        </xs:sequence>
      </xs:complexType>
    </xs:element>
    <xs:element name="CompositionTimecode" type="cpl:CompositionTimecodeType"
                    minOccurs="0" />
    <xs:element name="EditRate" type="dcml:RationalType" />
    <xs:element name="TotalRunningTime" minOccurs="0">
```

```
      <xs:simpleType>
        <xs:restriction base="xs:string">
          <xs:pattern value="[0-9][0-9]:[0-5][0-9]:[0-5][0-9]" />
        </xs:restriction>
      </xs:simpleType>
    </xs:element>
    <xs:element name="LocaleList" minOccurs="0">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="Locale" type="cpl:LocaleType" maxOccurs="unbounded"/>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
    <xs:element name="ExtensionProperties" minOccurs="0">
      <xs:complexType>
        <xs:sequence>
          <xs:any namespace="##other" processContents="lax" maxOccurs="unbounded"/>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
    <xs:element name="SegmentList">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="Segment" type="cpl:SegmentType" maxOccurs="unbounded"/>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
    <xs:element name="Signer" type="ds:KeyInfoType" minOccurs="0" />
    <xs:element ref="ds:Signature" minOccurs="0" />
  </xs:sequence>
</xs:complexType>
```

6.1.1 Id

The Id element shall uniquely identify the Composition Playlist instance. Any two Composition Playlist instances may have identical Id values if and only if the two Composition Playlist instances are identical.

6.1.2 Annotation

The Annotation element shall be a free-form, human-readable annotation describing the composition. It is meant strictly as a display hint to the user.

6.1.3 IssueDate

The IssueDate element shall indicate the time and date at which the Composition Playlist was issued.

6.1.4 Issuer

The Issuer element shall be a free-form, human-readable annotation that identifies the entity that created the Composition Playlist. It is meant strictly for display to the user.

Note: The Signer element defined in Section 6.1.17 is used to identify the entity that digitally signed the Composition Playlist.

6.1.5 Creator

The Creator element shall be a free-form, human-readable annotation that identifies the device or software program used to create the Composition Playlist, the facility that created the Composition Playlist and the operator that created the Composition Playlist. It is meant strictly for display to the user.

6.1.6 ContentOriginator

The ContentOriginator element shall be a free-form, human-readable annotation that identifies the originator of the content underlying the composition, as opposed to the Issuer, which identifies the entity that created the Composition Playlist. It is meant strictly for display to the user.

6.1.7 ContentTitle

The ContentTitle element shall contain a human-readable title for the composition, e.g. *The Jazz Singer*. It is strictly meant as a display hint to the user.

6.1.8 ContentKind

The ContentKind element shall be human-readable and indicate the kind of work represented by the Composition. ContentKindType is defined in Section 6.4.

The scope attribute of the ContentKind element shall determine the permissible values of the ContentKind element:

- if the scope attribute is absent from the ContentKind element, or set to its default value, the value of the ContentKind element shall match one of the values listed in Table 3;
- if the scope attribute is present and equal to "http://www.smpte-ra.org/schemas/2067-3/2016#content-kind", the value of the ContentKind element shall match one of the values listed in Table 4;
- otherwise, the value of the ContentKind element is outside the scope of this specification. In any case, this element is intended to be displayed to the user.

Table 3 – Permitted Content Kind values under the default scope

Kind	*Description*
advertisement	Content promoting a product or service other than an upcoming feature.
feature	A theatrical feature.
psa	Public service announcement.
rating	Slate/still image indicating the recommended age group permitted to view the content to follow. This rating is generally unique per country.
short	Non advertising/promotional content (3 to 15 minutes) typically before a theatrical feature.
teaser	Very short (typically less than 1 minute) content promoting an upcoming theatrical feature.
test	Content used to test, calibrate or setup equipment.
trailer	Short (2 to 3 minutes) content promoting an upcoming theatrical feature.
transitional	Extremely short content (1 to 15 seconds) separating unrelated compositions.
episode	Part of a dramatic work such as a serial television program.
highlights	Edited sequence of one or more memorable parts of an event, feature, episode or period of time.
event	Activity that happens at a given place and time, especially one of importance, including a music performance, a contest making up a sports competition.

Table 4 – Additional Content Kind values

Kind	*Description*
supplemental	Content created as extra material for a feature or episode such as behind the scenes material or deleted scenes.
documentary	A program that contains factual information based on real world events and/or people.

6.1.9 ContentVersionList

All ContentVersion elements within the ContentVersionList element shall be synonyms that uniquely identify the content represented by the Composition Playlist. This is in contrast with the Id element, specified in Section 6.1.1 , which identifies a particular instance of the Composition Playlist.

No two ContentVersion elements shall have identical Id elements.

Two Composition Playlist instances shall be assumed to refer to the same content if they have in common at least one Id element of a ContentVersion element. This specification makes no assumption of the definition of "same content", which is left to the issuer of the Composition Playlist instance.

The <any> element may be used to extend ContentVersion elements. Implementations shall ignore any such elements from a namespace it does not recognize.

ContentVersionType is defined in Section 6.8.

6.1.10 EssenceDescriptorList

The information provided by the EssenceDescriptorList element is intended to assist users in scheduling and tracking content. In case of a conflict between an EssenceDescriptor element and descriptive information in the underlying Track File, the latter shall take precedence. Implementations should not rely on this information for transcoding or playout.

6.1.10.1 EssenceDescriptor

Each EssenceDescriptor element of the EssenceDescriptorList element shall be a human-readable representation of selected descriptive information present in one or more Track Files. Collectively these EssenceDescriptor elements describe the essence referred to by the composition.

Each EssenceDescriptor shall be referenced through its Id element by at least one Resource of type derived from TrackFileResource — see Section 6.12.1.

For example, given a composition referencing one stereo and one 5.1 audio Track File, there would be at least two audio EssenceDescriptor elements. Each of the Resources referencing either of two Track Files would reference the corresponding EssenceDescriptor element through the SourceEncoding element.

The defining specification of each Track File referenced by the Composition Playlist shall specify the content of its associated EssenceDescriptor element and the process by which it is populated. Specifically, it shall specify the elements, if any, that are inserted in lieu of the placeholder <any> element. For example, the EssenceDescriptor element for a Track File using the MXF file format could consist of an XML representation of its File Descriptor structure using SMPTE ST 434.

The EssenceDescriptorBaseType is specified in Section 6.7.

6.1.11 CompositionTimecode

The CompositionTimecode element provides the information necessary to associate a canonical synthetic timecode with the Composition Playlist timeline – see Section 8. CompositionTimecodeType is defined in Section 6.2.

6.1.12 EditRate

The EditRate element shall define the Composition Edit Rate. The duration of a Composition Edit Unit shall be the multiplicative inverse of Composition Edit Rate, i.e. 1/(Composition Edit Rate).

6.1.13 TotalRunningTime

The TotalRunningTime element shall indicate the approximate duration of the Composition as hours:minutes:seconds at the time when Composition Playlist was issued.

This information is provided for informational purposes only, to assist users in scheduling and tracking content. Implementations should not rely on this information for transcoding or playout.

The exact running time of the Composition shall be the duration of the Composition as calculated in Section 7.1, divided by the Composition Edit Rate (Section 6.1.12), and shall always take precedence in case of conflict.

6.1.14 LocaleList

Each Locale element contained in the LocaleList element shall specify the intended audience of the Composition, e.g. in terms of region, language and maturity rating.

6.1.15 ExtensionProperties

The ExtensionProperties element contains an unordered list of <any> elements, which may be used by applications to add descriptive metadata to the Composition Playlist, e.g. a reference to a source EDL.

Implementations shall ignore any children of ExtensionProperties belonging to a namespace they do not recognize.

6.1.16 SegmentList

The SegmentList element shall contain an ordered list of Segment elements of type SegmentType. SegmentType is defined in Section 6.9.

Segments are reproduced sequentially, as specified in Section 7.2.

6.1.17 Signer

The Signer element shall uniquely identify the entity that digitally signed the Composition Playlist. If the Signer element is present, then the Signature element shall also be present.

If X.509 certificates are used as specified in [XML Digital Signature], then the Signer element shall contain one X509Data element containing one X509IssuerSerial element, which uniquely identifies the certificate used to sign the Composition Playlist.

6.1.18 Signature

The Signature element shall contain a digital signature authenticating the Composition Playlist.

If the Signature element is present, then the Signer element shall be present.

The digital signature shall be enveloped, as specified in [XML Digital Signature], and apply to the entire Composition Playlist. The signature is generated by the signer, as identified by the Signer element.

6.2 CompositionTimecodeType

Table 5 – CompositionTimecodeType schema definition

```
 <xs:complexType name="CompositionTimecodeType">
  <xs:sequence>
  <xs:element name="TimecodeDropFrame" type="xs:boolean" />
  <xs:element name="TimecodeRate" type="xs:positiveInteger" />
  <xs:element name="TimecodeStartAddress" type="cpl:TimecodeType" />
  </xs:sequence>
 </xs:complexType>
```

6.2.1.1 TimecodeDropFrame

The TimecodeDropFrame element shall indicate if the timecode is drop frame (value is true) or non-drop frame (value is false).

6.2.1.2 TimecodeRate

The TimecodeRate element shall specify the nearest integer frames per second rate of the timecode, e.g. 24, 30, 25…

6.2.1.3 TimecodeStartAddress

The TimecodeStartAddress shall specify the value of the timecode at the beginning of the Composition.

TimecodeType is specified in Section 6.3.

The choice of delimiters used in the value of TimecodeStartAddress has no effect on the canonical synthetic timecode. In particular, as specified in Section 8, only the value of the TimeCodeDropFrame element is used to determine whether the canonical synthetic timecode is drop-frame or non-drop-frame and only the value of TimecodeRate is used to determine the frame rate.

6.3 TimecodeType

The TimecodeType represents a timecode value.

Table 6 – TimecodeType schema definition

```
<xs:simpleType name="TimecodeType">
 <xs:restriction base="xs:string">
  <xs:pattern value="[0-2][0-9](:|/|;|,|\.|\+|\-)[0-5][0-9](:|/|;|,|\.|\+|\-)[0-5][0-
                  9](:|/|;|,|\.|\+|\-)[0-5][0-9]" />
 </xs:restriction>
 </xs:simpleType>
```

The value of a TimeCodeType instance consists of a sequence of integers separated by delimiters – see Table 6. Starting from the left of the value,

- the first integer shall be the hours field of the timecode;
- the second integer shall be the minutes field of the timecode value;

- the third integer shall be the seconds field of the timecode value; and
- the fourth integer shall be the frames field of the timecode value.

6.4 ContentKindType

The ContentKindType represents the kind of content underlying the composition.

Table 7 – ContentKindType schema definition

```
  <xs:complexType name="ContentKindType">
    <xs:simpleContent>
      <xs:extension base="xs:string">
        <xs:attribute name="scope" type="xs:anyURI" use="optional"
                      default="http://www.smpte-ra.org/schemas/2067-3/2013#content-
                    kind" />
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>
```

6.5 LocaleType

The LocaleType contains information specific to the locale(s) to which the Composition is destined.

Table 8 – LocaleType schema definition

```
<xs:complexType name="LocaleType">
  <xs:sequence>
    <xs:element name="Annotation" type="dcml:UserTextType" minOccurs="0" />
    <xs:element name="LanguageList" minOccurs="0">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="Language" type="xs:string" maxOccurs="unbounded" />
        </xs:sequence>
      </xs:complexType>
    </xs:element>
    <xs:element name="RegionList" minOccurs="0">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="Region" type="xs:string" maxOccurs="unbounded" />
        </xs:sequence>
      </xs:complexType>
    </xs:element>
    <xs:element name="ContentMaturityRatingList" minOccurs="0">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="ContentMaturityRating"
                    type="cpl:ContentMaturityRatingType" maxOccurs="unbounded"/>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
  </xs:sequence>
</xs:complexType>
```

6.5.1 Annotation

The Annotation element shall be a free-form, human-readable annotation describing the Locale. It is meant strictly as a display hint to the user.

6.5.2 LanguageList

Each Language element within the LanguageList element shall represent a language (spoken and/or written) of the audience in the Locale and shall be a Language Tag as specified in [RFC 5646].

The Language element values are not necessarily equal to the spoken and/or written language of the underlying essence.

6.5.3 RegionList

Each Region element of the RegionList element shall represent a region in the Locale and shall be a valid region subtag of the Language Subtag Registry specified in [RFC 5646].

Note: As of the publication date of this specification, the set of region subtags in the Language Subtag Registry consists of a subset of the geographical regions specified in ISO 3166-1 and U.N. M.49. The Language Subtag Registry is maintained by IANA at http://www.iana.org/assignments/language-subtag-registry and is the definitive list of region subtags.

6.5.4 ContentMaturityRatingList

Each ContentMaturityRating element of the ContentMaturityRatingList element shall contain a content maturity rating associated with the Locale.

There shall be only one ContentMaturityRating element with a given value of the Agency element.

ContentMaturityRatingType is defined in Section 6.6.

Note: Content maturity rating indicates suitability for audiences in terms of issues such as sex, violence, substance abuse, profanity, impudence or other types of mature content. It is distinct from audience ratings and critical ratings.

6.6 ContentMaturityRatingType

Table 9 – ContentMaturityRatingType schema definition

```
<xs:complexType name="ContentMaturityRatingType">
  <xs:sequence>
    <xs:element name="Agency" type="xs:anyURI" />
    <xs:element name="Rating" type="xs:string" />
    <xs:element name="Audience" minOccurs="0" >
      <xs:complexType>
        <xs:simpleContent>
          <xs:extension base="xs:string">
            <xs:attribute name="scope" type="xs:anyURI" use="required"/>
          </xs:extension>
        </xs:simpleContent>
      </xs:complexType>
    </xs:element>
    <xs:any namespace="##other" processContents="lax" minOccurs="0"
                  maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>
```

The <xs:any> element allows information, beyond the textual representation of the rating contained in the Rating element, to be associated with a rating. Such information can contain, for instance, specific advisories or other information specific to the Agency issuing the rating.

6.6.1 Agency

The Agency element shall uniquely identify the agency issuing the rating. Assignment, registration and listing of Agency values are beyond the scope of this standard.

6.6.2 Rating

The Rating element shall contain a human-readable representation of the rating, which is intended to be displayed to the user.

For each unique agency, there should be a number of permissible Ratings values. Assignment, registration and listing of defined Label values per Agency value is beyond the scope of this standard.

Table 10 provides an illustrative combination of Agency and corresponding Rating values.

Table 10 – Example Ratings (Informative)

Agency	*Permitted Rating Values*
http://agency.example.invalid/2011-ratings	R, PG, PG-13, G, NC-17

6.6.2.1 Audience

The Audience element shall contain a human-readable representation of the intended the target audience of the Composition.

The scope attribute of the Audience element shall determine the permissible values of the Audience element. Specific values of Audience element's scope attribute and corresponding permissible values of the Audience element are locale-specific and beyond the scope of this specification. In any case, the value of the Audience element is intended to be displayed to the user.

Table 11 provides an illustrative combination of scope and permitted Audience values.

Table 11 – Example Audience values (Informative)

Scope	*Permitted values*
http://scope.example.invalid/2011-audience	Adult, Mature, General Family, Teen, Children

6.7 EssenceDescriptorBaseType

Table 12 – EssenceDescriptorBaseType schema definition

```
<xs:complexType name="EssenceDescriptorBaseType">
 <xs:sequence>
  <xs:element name="Id" type="dcml:UUIDType" />
  <xs:any namespace="##other" processContents="lax" minOccurs="0"
                  maxOccurs="unbounded"/>
 </xs:sequence>
 </xs:complexType>
```

6.7.1 Id

The Id element shall uniquely identify the EssenceDescriptor element.

6.8 ContentVersionType

Table 13 – ContentVersionType schema definition

```
<xs:complexType name="ContentVersionType">
  <xs:sequence>
    <xs:element name="Id" type="xs:anyURI" />
  <xs:element name="LabelText" type="dcml:UserTextType" />
  <xs:any namespace="##other" processContents="lax" minOccurs="0"
                  maxOccurs="unbounded"/>
 </xs:sequence>
 </xs:complexType>
```

6.8.1 Id

The Id element shall identify the content represented by the Composition Playlist.

6.8.2 LabelText

The LabelText element shall be a human-readable description of the content, e.g. “French (1.85 picture, 16.1 sound, dubbed)”, associated with the composition.

For example, two compositions can have identical ContentTitle value and yet refer to two different versions and therefore have distinct LabelText values, such as "French (dubbed)" and "French (original)".

6.9 SegmentType

Table 14 – SegmentType schema definition

```
<xs:complexType name="SegmentType">
 <xs:sequence>
  <xs:element name="Id" type="dcml:UUIDType"/>
  <xs:element name="Annotation" type="dcml:UserTextType" minOccurs="0" />
  <xs:element name="SequenceList">
  <xs:complexType>
   <xs:sequence>
   <xs:element name="MarkerSequence" type="cpl:SequenceType" minOccurs="0" />
   <xs:any namespace="##other" processContents="lax" minOccurs="0"
                  maxOccurs="unbounded" />
   </xs:sequence>
  </xs:complexType>
  </xs:element>
 </xs:sequence>
 </xs:complexType>
```

6.9.1 Id

The Id element shall uniquely identify the Segment for asset management purposes.

6.9.2 Annotation

The Annotation element shall be a free-form, human-readable annotation describing the Segment. It is meant strictly as a display hint to the user.

6.9.3 SequenceList

Each element in SequenceList is called a Sequence and shall be derived, directly or indirectly, from SequenceType, which is defined in Section 6.10.

Each Sequence contains a TrackId element that links Sequences across Segment boundaries, creating a Virtual Track that spans the entire Composition Playlist. All Sequences with equal TrackId shall belong to the same Virtual Track. A given TrackId value shall be used by only one Sequence in each Segment and, if used in one Segment, it shall be used by exactly one Sequence in all other Segments.

Each Sequence and Virtual Track shall be associated with a single aspect of the presentation and therefore a single kind of essence. As specified in Section 7.2, Sequences within a Segment are reproduced in parallel.

In order to remain agnostic to the details of the essence, this specification does not define Sequence elements specific to particular essence kinds, e.g. audio and image. Instead the <any> element may be used to add additional Extension Sequences. All Extension Sequences shall be derived, directly or indirectly, from SequenceType. An implementation shall ignore any Extension Sequence from a namespace it does not recognize.

Extension Sequences should have unique, descriptive names and should appear only once in a given Segment. Extension specifications that allow multiple instances of an element in a Segment should provide both a means of differentiating instances within a Segment.

This specification defines a single essence-agnostic Sequence element, the MarkerSequence element, which is specified in Section 6.9.3.1.

6.9.3.1 MarkerSequence

The MarkerSequence element defines markers – see Section 6.14.

The MarkerSequence element shall contain Resources that are instances or subclasses of type MarkerResourceType.

6.10 SequenceType

Table 15 – SequenceType schema definition

```
<xs:complexType name="SequenceType">
 <xs:sequence>
  <xs:element name="Id" type="dcml:UUIDType"/>
  <xs:element name="TrackId" type="dcml:UUIDType"/>
  <xs:element name="ResourceList">
     <xs:complexType>
       <xs:sequence>
         <xs:element name="Resource" type="cpl:BaseResourceType"
maxOccurs="unbounded"/>
       </xs:sequence>
     </xs:complexType>
   </xs:element>
 </xs:sequence>
 </xs:complexType>
```

A Sequence consists of a list of Resources that are played sequentially —– see Section 7.3.

All Resources elements within a Sequence shall be of the same type, as defined in [XML Schema Part 1: Structures].

Resources can reference external Assets, including Track Files containing image, audio, data, etc., or can store their Asset within the Composition Playlist structure, e.g. Marker Resources.

6.10.1 Id

The Id element shall uniquely identify the Sequence.

6.10.2 TrackId

The TrackId shall uniquely identify the Virtual Track to which the Sequence belongs.

6.10.3 ResourceList

The elements of ResourceList shall be ordered. All Resource elements share common attributes and are specified as a set of types derived from a common structure, namely the BaseResourceType, defined in Section 6.11.

6.11 BaseResourceType

Table 16 – BaseResourceType schema definition

```
<xs:complexType name="BaseResourceType" abstract="1">
 <xs:sequence>
  <xs:element name="Id" type="dcml:UUIDType" />
  <xs:element name="Annotation" type="dcml:UserTextType" minOccurs="0" />
  <xs:element name="EditRate" type="dcml:RationalType" minOccurs="0" />
  <xs:element name="IntrinsicDuration" type="xs:nonNegativeInteger" />
  <xs:element name="EntryPoint" type="xs:nonNegativeInteger" minOccurs="0" />
  <xs:element name="SourceDuration" type="xs:nonNegativeInteger" minOccurs="0" />
  <xs:element name="RepeatCount" type="xs:positiveInteger" minOccurs="0" />
</xs:sequence>
 </xs:complexType>
```

BaseResourceType is the abstract base class for Resources. A Resource selects a portion an underlying Asset for reproduction — see Section 7.4.

6.11.1 Id

The Id element shall uniquely identify this specific Resource instance. The means of identifying the underlying Asset is left to subclasses.

6.11.2 Annotation

The Annotation element shall be a free-form, human-readable annotation describing the Resource. It is meant strictly as a display hint to the user.

6.11.3 EditRate

The EditRate element shall define the Edit Rate of the Resource, i.e. the Resource Edit Rate. The duration of a Resource Edit Unit shall be the multiplicative inverse of the Resource Edit Rate, i.e. 1/(Resource Edit Rate).

If the EditRate element is absent, the Edit Rate of the Resource shall be equal to the Composition Edit Rate.

6.11.4 IntrinsicDuration

The IntrinsicDuration element shall be the native duration of the underlying Asset in Resource Edit Units. It shall not take into account the values of the EntryPoint and SourceDuration elements.

The defining specification BaseResourceType subclasses shall specify the native duration of the Asset.

6.11.5 EntryPoint

The EntryPoint element shall identify the start of the playable region as a time offset from the native start point of the underlying Asset in Resource Edit Units. If the EntryPoint element is absent, a value of 0 shall be assumed.

The defining specification of BaseResourceType subclasses shall specify the native start point of the Asset.

6.11.6 SourceDuration

The SourceDuration element shall define the duration of the playable region in Resource Edit Units.

If present, it shall be between 0 (zero) and IntrinsicDuration – EntryPoint.

If absent, it shall be equal to IntrinsicDuration – EntryPoint.

6.11.7 RepeatCount

The RepeatCount element shall define the number of times the repeat playable region is repeated during playback. The repeat count shall be equal to one if the RepeatCount element is absent.

6.12 TrackFileResourceType

Table 17 – TrackFileResourceType schema definition

```
<xs:complexType name="TrackFileResourceType">
  <xs:complexContent>
    <xs:extension base="cpl:BaseResourceType">
      <xs:sequence>
        <xs:element name="SourceEncoding" type="dcml:UUIDType" />
        <xs:element name="TrackFileId" type="dcml:UUIDType" />
        <xs:element name="KeyId" type="dcml:UUIDType" minOccurs="0" />
        <xs:element name="Hash" type="xs:base64Binary" minOccurs="0" />
       <xs:element name="HashAlgorithm" type="ds:DigestMethodType" minOccurs="0"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

The TrackFileAssetType is a Resource whose underlying Asset is an external Track File.

The defining specification for each Track File referenced in a Composition Playlist shall specify its native duration and native start point.

6.12.1 SourceEncoding

The SourceEncoding element shall reference one element of the EssenceDescriptorList through its Id element.

6.12.2 TrackFileId

The TrackFileId shall uniquely identify the underlying Track File.

The defining specification for each Track File referenced in a Composition Playlist shall specify the identifier for use with TrackFileId.

Mapping of UUID values to actual Track File locations is beyond the scope of this document. See Annex D.9 for additional details.

6.12.3 KeyId

The KeyId element shall uniquely identify the cryptographic key used to encrypt the underlying Track File. It shall be present if any portion of the underlying track file is encrypted.

The mapping of key identifiers to actual key values is beyond the scope of this document. See Annex D.10 for additional details.

6.12.4 Hash

The Hash element shall contain the hash (message digest) of the underlying track file. It shall be computed by applying the algorithm specified by the HashAlgorithm element over the entire Track File and encoding the result using Base64 representation [RFC 2045].

Note: The Hash element is intended to be used to confirm Track File integrity, including in the context of security applications, and therefore reflects any change, however minute, to the Track File.

6.12.5 HashAlgorithm

As detailed in [XML Digital Signature], the HashAlgorithm element specifies the algorithm used to compute the Hash element.

Implementations shall support the algorithms listed in Table 18.

Table 18 – Hash Algorithms

HashAlgorithm	Defining specification
http://www.w3.org/2000/09/xmldsig#sha1	[XML Digital Signature]

The HashAlgorithm element shall be present if the Hash algorithm is present, and shall be absent otherwise.

6.13 MarkerResourceType

Table 19 – MarkerResourceType schema definition

```
<xs:complexType name="MarkerResourceType">
  <xs:complexContent>
    <xs:extension base="cpl:BaseResourceType">
      <xs:sequence>
        <xs:element name="Marker" type="cpl:MarkerType" maxOccurs="unbounded"
                    minOccurs="0"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

The MarkerResourceType contains content markers, e.g. First Frame of End Credits. Each content marker shall be represented by a Marker element of type MarkerType, specified in Section 6.14. Content markers may be repeated as there will likely be multiple instances of several types of content segments throughout a given program (i.e. production logos, commercial blacks, etc).

The Marker Resource defines a timeline where the position of each Marker is determined by its offset from the start of the timeline.

The native start point of a MarkerResourceType instance shall be the start of the timeline, i.e. correspond to an offset of 0 on the timeline.

The native duration of a MarkerResourceType instance, as indicated by the IntrinsicDuration element, shall be set to any value equal or larger to the largest Offset value within all its Marker elements.

The edit rate of a MarkerResourceType instance may be set to any value. It is usually set to the Composition Edit Rate.

6.14 MarkerType

Table 20 – MarkerType schema definition

```
<xs:complexType name="MarkerType">
 <xs:sequence>
   <xs:element name="Annotation" type="dcml:UserTextType" minOccurs="0" />
   <xs:element name="Label">
     <xs:complexType>
       <xs:simpleContent>
         <xs:extension base="xs:string">
           <xs:attribute name="scope" type="xs:anyURI" use="optional"
                default="http://www.smpte-ra.org/schemas/2067-3/2013#standard-
                markers" />
         </xs:extension>
       </xs:simpleContent>
     </xs:complexType>
   </xs:element>
   <xs:element name="Offset" type="xs:nonNegativeInteger"/>
 </xs:sequence>
</xs:complexType>
```

6.14.1.1 Annotation

The Annotation element shall be a free-form, human-readable annotation describing the composition. It is meant strictly as a display hint to the user.

6.14.1.2 Label

The Label element shall contain a textual representation of the marker.

The scope attribute of the Label element shall determine the permissible values of the Label element.

The combination of the scope attribute and the value of the Label element determines the meaning of the marker.

Table 21 specifies values of the scope attribute and corresponding Label element that may be used.

Table 21 – Marker Labels

Label	*Description*
scope="http://www.smpte-ra.org/schemas/2067-3/2013#standard-markers"	
FFBT	First Frame of Bars and Tone
FFCB	First Frame of Commercial Blacks
FFCL	First Frame of Company/Production Logo
FFDL	First Frame of Distribution Logo
FFEC	First Frame of End Credits. First displayable frame of content that contains any intensity of the End Credits (a non zero alpha value), which appear at the end of a feature.
FFHS	First Frame of Head Slate
FFMC	First displayable frame of content that contains any intensity of moving, rolling or scrolling credits (a non-zero alpha value), which appear at the end of the feature.
FFOB	First Frame of Ratings Band. First displayable frame of content of the Rating Band, which is usually a slate at the beginning of a feature.
FFOC	First Frame of Composition. The first frame of a composition that is intended for display.
FFOI	First Frame of Intermission.
FFSP	First Frame of Digital Sync Pop
FFTC	First Frame of Title Credits. First displayable frame of content that contains any intensity of the Title Credits (a non zero alpha value), which appear at the beginning of a feature.
FFTS	First Frame of Tail Slate
FTXC	First Frame of Textless Title Credits
FTXE	First Frame of Textless End Credits
FTXM	First Frame of Textless Material Segment
LFBT	Last Frame of Bars and Tone
LFCB	Last Frame of Commercial Blacks
LFCL	Last Frame of Company/Production Logo
LFDL	Last Frame of Distribution Logo
LFEC	Last Frame of End Credits. Last displayable frame of content that contains any intensity of the End Credits (a non zero alpha value), which appear at the end of a feature.
LFHS	Last Frame of Head Slate
LFMC	Last displayable frame of content that contains any intensity of moving, rolling or scrolling credits (a non-zero alpha value), which appear at the end of the feature.
LFOB	Last Frame of Ratings Band. Last displayable frame of content of the Rating Band, which is usually a slate at the beginning of a feature.

LFOC	Last Frame of Composition. The last frame of a composition that is intended for display.
LFOI	Last Frame of Intermission.
LFSP	Last Frame of Digital Sync Pop
LFTC	Last Frame of Title Credits. Last displayable frame of content that contains any intensity of the Title Credits (a non zero alpha value), which appear at the beginning of a feature.
LFTS	Last Frame of Tail Slate
LTXC	Last frame of Textless Title Credits
LTXE	Last Frame of Textless End Credits
LTXM	Last frame of Textless Material Segment
FPCI	Fixed Point Candidate Insertion. Indicates possible point in the timeline where it would be allowable to insert content downstrean. This is for material that may not have commercial blacks, but could indicate a candidate point where a commercial could be inserted.
FFCO	First Frame of Candidate Overlay. First frame of a sequence of frames where overlays, e.g. commercial overlays, may be placed.
LFCO	Last Frame of Candidate Overlay. Last frame of a sequence of frames where overlays, e.g. commercial overlays, may be placed.
FFOA	Audio First Frame. First frame of audio ring-in/ring-out where the video is in black.
LFOA	Audio Last Frame. Last frame of audio ring-in/ring-out where the video is in black.
scope="http://www.smpte-ra.org/schemas/2067-3/2016#standard-markers"	
FFDC	First Frame of Dubbing Credits. First displayable frame of content that contains any intensity of dubbing credits
LFDC	Last Frame of Dubbing credits: Last displayable frame of content that contains any intensity of dubbing credits

Other values of the scope attribute may be present, but are outside the scope of this specification.

In any case, the value of the element, or its associated description if available, is intended to be displayed to the user.

6.14.1.3 Offset

The Offset element shall define the position of the marker from the start of the Marker Resource timeline. It shall be expressed in Resource Edit Units.

7 Timeline and Synchronization Model

The following specifies the Composition timeline, which allows synchronized processing of the essence contained in underlying Assets, including playback and transcoding.

7.1 Composition

The Composition timeline shall consist of the concatenation, without gaps, of the timeline of all its Segments in the order they appear in the SegmentList element – see Figure 2.

The temporal offset T_k from the start of Segment k shall correspond to temporal offset $T = T_k + \Sigma_{j=1}^{k-1} D_j$ from the start of the Composition, where D_j is the duration of the j^{th} Segment.

The duration of the Composition shall be the sum of the duration of its Segments.

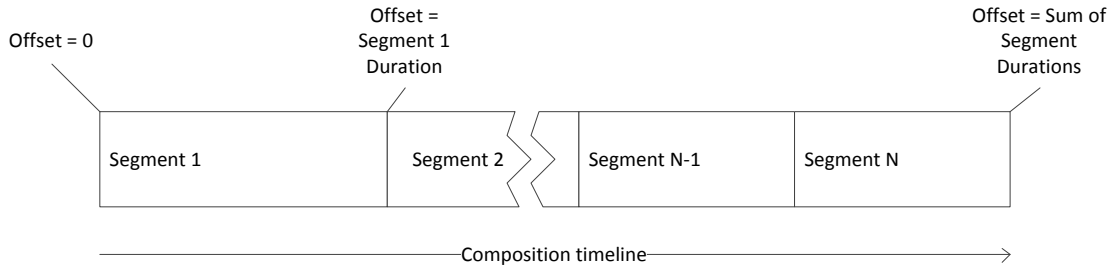

Figure 2 – Composition Playlist Timeline

7.2 Segment

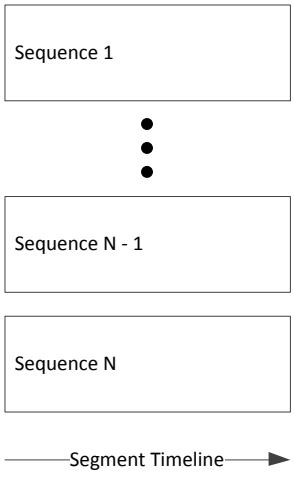

Figure 3 – Segment Timeline

The timeline of a Segment shall consist of the timeline of all Sequences within SequenceList arranged in parallel, with their start and end points aligned — see Figure 3.

The temporal offset T' from the start of the Segment shall correspond to temporal offset T within each Sequence.

The duration of the Segment shall be equal to the duration of its Sequences and all Sequences within a Segment shall have the same duration.

7.3 Sequence

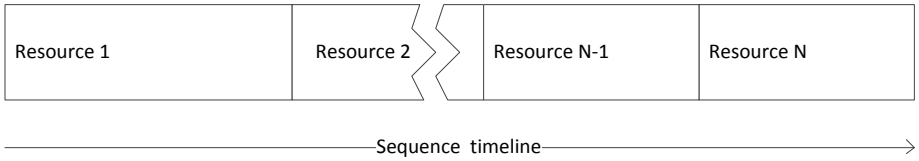

Figure 4 – Sequence Timeline

The timeline of a Sequence shall consist of the concatenation, without gaps, of the timeline of all its Resources in the order they appear in the ResourceList element — see Figure 4.

The temporal offset T_k from the start of Resource k shall correspond to temporal offset $T = T_k + \Sigma_{j=1}^{k-1} D_j$ from the start of the Sequence, where D_j is the duration of the j^{th} Resource.

The duration of a Sequence shall be the sum of the duration of its Resources and shall be an integer number of Composition Edit Units.

Note: Not all underlying Track Files within a Segment necessarily have the same duration since Resources can reference only a portion of the underlying Track File and Sequences can contain multiple Resources.

7.4 Resource

As depicted in Figure 5, the Resource timeline shall consist of the playable region of the underlying Asset repeated RepeatCount times, without gaps.

The playable region shall start exactly at a temporal offset EntryPoint and stop exactly at a temporal offset EntryPoint + SourceDuration from the native start point of the underlying asset.

Note: An application can allow the playable region start or end within an essence sample, e.g. video frame, audio sample or coded audio frame. In such an application, the duration of the playable region of a 48 kHz audio asset could, for instance, be set to exactly 3 image frames sampled at 30,000/1,001 Hz, or 1,016.6 audio samples. The behavior of implementations when reproducing a playable region containing a fractional number of essence samples is left to the application.

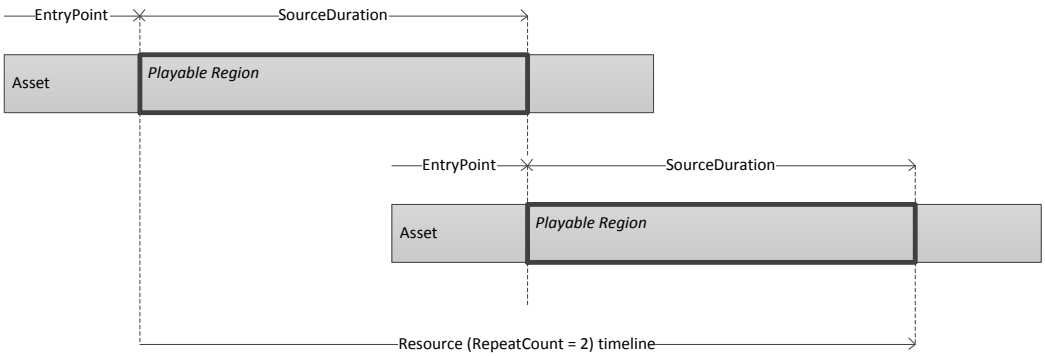

Figure 5 – Resource Timeline

The duration of a Resource shall be SourceDuration times RepeatCount.

The temporal offset T_k from the start of playable region k, where $1 \leq k \leq$ RepeatCount, shall correspond to temporal offset $T = T_k + (k - 1) D$ from the start of the first playable region, where D is the Source Duration.

Assets shall be in editorial sync. In other words, the Composition timeline shall not compensate for any processing delay that occurs during processing, e.g. playback or transcoding.

The Composition Playlist does not define any form of transition or overlap between playable regions: the reproduction of one playable region ends immediately when the following one starts, i.e. a "butt-edit". As such, the essence contained in the underlying assets should be prepared for artifact-free reproduction across these boundaries. For example, the boundary between the playable regions of two audio assets should be crafted to ensure a continuous waveform or should occur in an area of digital silence or at a zero crossing.

8 Canonical Synthetic Timecode

While the Composition Playlist specifies precise essence and metadata synchronization within the Composition, it is not always sufficient for the synchronization of downstream devices and processes. The following specifies a canonical synthetic timecode as specified in [ST 12-1], for values of the TimecodeRate element less than or equal to 60. This specification does not specify a canonical synthetic timecode for values of the TimecodeRate element greater than 60.

The timecode frame rate shall be equal to the TimecodeRate element value.

The starting timecode value shall be equal to the TimecodeStartAddress element value and shall correspond to time offset 0 within the Composition timeline.

The timecode drop frame flag shall be set to the TimecodeDropFrame element value.

If the TimecodeRate value is less than or equal to 30, the timecode shall increment for every Composition Edit Unit duration, subject to the rules in [SMPTE ST 12-1] regarding dropframe and non-dropframe counts.

If the TimecodeRate value is larger than 30 and less or equal to 60, the timecode shall increment for every other Composition Edit Unit duration. Where the timecode is conveyed as VITC data (for example as in ATC), the field mark flag of the VITC shall be set to zero for the first frame of a pair and to one for the second frame of a pair.

This specification specifies only the values of hours, minutes, seconds and frames of the canonical synthetic timecode. It does not specify the value of SMPTE ST 12-1 color frame flag, binary groups and binary group flags.

Annex A Bibliography (Informative)

SMPTE ST 377-1:2011, Material Exchange Format (MXF) — File Format Specification

[1)]SMPTE ST 434:2006, Material Exchange Format — XML Encoding for Metadata and File Structure Information

SMPTE ST 429-7:2006 D-Cinema Packaging — Composition Playlist

International Organization for Standardization, ISO 3166-1:2006. Codes for the representation of names of countries and their subdivisions — Part 1: Country codes, November 2006

Statistics Division, United Nations. Standard Country or Area Codes for Statistical Use, Revision 4 (United Nations publication, Sales No. 98.XVII.9, June 1999

[1)] **Note:** The reference to the superseded version of this standard is intentional. The current version was not reviewed by the Technical Committee to verify if it is appropriate for use with this document.

.

Annex B Consolidated Schema (Informative)

This specification is accompanied by the following element, which is an XML schema document as specified in [XML Schema Part 1: Structures].

imf-cpl-20160411.xsd

This element collects the XML schema definitions defined in this specification. It is informative and, in case of conflict, this specification takes precedence.

Annex C Consideration for Track File Specifications (Informative)

For convenience, the following highlights requirements for the defining specification of Track Files that are referenced from the Composition Playlist.

TrackFileId. See Section 6.12.2.

Native Duration. See Section 6.12.

Native Start Point. See Section 6.12.

EssenceDescriptor. See Section 6.1.10.

Annex D Considerations for Applications (Informative)

The following highlights areas of the Composition Playlist that applications can constrain, specify and extend to satisfy their specific needs.

D.1 Content Version

Applications can support only specific kinds of Content Version Ids, e.g. UUID, ISAN, Entertainment ID Registry (EIDR), etc.

D.2 EditRate

Applications can constrain the Composition Edit Rate to specific values or range of values, e.g. to match permitted frame rates of the underlying image essence.

D.3 Timeline

Applications can constrain the Composition timeline to fit their needs, e.g. define a minimum Segment duration or specify that the Segment boundaries match an underlying essence feature, e.g. commercial breaks.

D.4 Splice Points

As illustrated in Section 7.4, applications can specify reproduction behavior at playable region boundaries in the event that the playable region does not contains an integer number of essence samples, e.g. audio sample, video frame or coded audio frame...

D.5 Sequences

Applications can constrain the SequenceList to fit their needs, e.g. specify the kind and cardinality of sequences.

D.6 Entry Points and Source Durations

Depending on the nature of the underlying essence permitted by the application, an application can restrict the entry points and duration in Resources to ensure that playable regions contain an integer number of essence samples, e.g. audio sample, video frame or coded audio frame.

D.7 Track File characteristics

The characteristics of Track Files will typically be constrained, e.g. to a specific operational pattern in the case of MXF files.

D.8 Essence characteristics

Applications can require that all essence within a Virtual Track share the same characteristics, such as color space in the case of image or sample rate in the case of audio.

D.9 Mapping of TrackFileId values to actual Track File locations

Applications can define one or more means to map the TrackFileId values to the location of the underlying Track File. This can involve, for instance, a separate manifest file that associates a TrackFileId with a file system path.

D.10 Mapping of KeyId values to cryptographic keys

Applications can define one or more means to map the KeyId values to cryptographic keys.

D.11 Markers

As detailed in Section 6.14.1.1, an application can introduce additional markers without risk of collision with markers defined by other applications. This is done by defining any number of values for the Label element and associated them with a globally unique URI, which is used as the scope attribute of the same element.

Applications can further specify processing behavior based on the presence or absence of specific markers.

D.12 Digital Signature

Applications can constrain the algorithms and digital certificates that are used when generating Signature elements.

D.13 Minimum and Maximum Track File Duration

Applications can specify minimum and maximum values for Track File durations.

D.14 ExtensionProperties

As specified in Section 6.1.15, applications can introduce descriptive metadata elements to the Composition Playlist.

D.15 EssenceDescriptors

As specified in Section 6.1.10, applications can specify the means by which descriptive metadata is extracted from assets and exposed in the EssenceDescriptors element.

D.16 Content Maturity Rating Agencies and Audience

As specified in Section 6.5.4, application can refer to or specify values for the Rating, Agency and Audience elements.

D.17 Sequences

As specified in Section 6.10.3, applications can define subclasses of SequenceType to introduce new kinds of Sequences.

D.18 Resource

As specified in Section 6.10.3, applications can define subclasses of BaseResourceType to introduce new kinds of Resources.

Annex E Sample Instance (Informative)

This specification is accompanied by the following element, which is an XML document that contains a sample instance of the CompositionPlaylist element specified in Section 5.

imf-cpl-sample-20160411.xml

This element is for illustration only, and is neither intended to capture current or future practice, nor exercise all normative language contained in this specification.