INTERNATIONAL STANDARD

ISO 26429-7

First edition
2008-09-01

Digital cinema (D-cinema) packaging —

Part 7: Composition playlist

Emballage du cinéma numérique (cinéma D) —

Partie 7: Liste d'écoute de composition

Reference number
ISO 26429-7:2008(E)

ISO copyright office
Case postale 56 • CH-1211 Geneva 20
Tel. + 41 22 749 01 11
Fax + 41 22 749 09 47
E-mail copyright@iso.org
Web www.iso.org

Published in Switzerland

Foreword

ISO (the International Organization for Standardization) is a worldwide federation of national standards bodies (ISO member bodies). The work of preparing International Standards is normally carried out through ISO technical committees. Each member body interested in a subject for which a technical committee has been established has the right to be represented on that committee. International organizations, governmental and non-governmental, in liaison with ISO, also take part in the work. ISO collaborates closely with the International Electrotechnical Commission (IEC) on all matters of electrotechnical standardization.

International Standards are drafted in accordance with the rules given in the ISO/IEC Directives, Part 2.

The main task of technical committees is to prepare International Standards. Draft International Standards adopted by the technical committees are circulated to the member bodies for voting. Publication as an International Standard requires approval by at least 75 % of the member bodies casting a vote.

ISO 26429-7 was prepared by the Society of Motion Picture and Television Engineers (as SMPTE 429-7-2006) and was adopted, under a special "fast-track procedure", by Technical Committee ISO/TC 36, *Cinematography*, in parallel with its approval by the ISO member bodies.

ISO 26429 consists of the following parts, under the general title *Digital cinema (D-cinema) packaging*:

— *Part 3: Sound and picture track file*

— *Part 4: MXF JPEG 2000 application*

— *Part 6: MXF track file essence encryption*

— *Part 7: Composition playlist*

Introduction

This International Standard comprises SMPTE 429-7-2006 and the following informative notes.

— Table 2 (Content Kind): Business arrangements by national practice determine what is inclusive in the definition of a “feature”.

— Table 4 (Standard Marker Labels): An additional marker called “First Frame Lights On” (FFLO), which could be used by the production company to indicate an appropriate position within the end credits to turn on the lights, is optionally available and has been proposed for incorporation into SMPTE 429-7.

The International Organization for Standardization (ISO) draws attention to the fact that it is claimed that compliance with this document may involve the use of a patent.

ISO takes no position concerning the evidence, validity and scope of this patent right.

The holder of this patent right has assured ISO that he is willing to negotiate licences under reasonable and non-discriminatory terms and conditions with applicants throughout the world. In this respect, the statement of the holder of this patent right is registered with ISO. Information may be obtained from:

Eastman Kodak Company
Intellectual Property Transactions
343 State Street
Rochester, NY 14650
USA

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights other than those identified above. ISO shall not be held responsible for identifying any or all such patent rights.

SMPTE 429-7-2006

SMPTE STANDARD

D-Cinema Packaging — Composition Playlist

Table of Contents

Approved
October 3, 2006

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 Administrative practices.

Proposed SMPTE Standard 429-7 was prepared by Technology Committee DC28.

1 Scope

This document specifies the Composition Playlist structure. The Composition Playlist is a self-contained representation of a single complete D-Cinema work, such as a motion picture, or a trailer, or an advertisement, etc. The Composition Playlist consists of an ordered sequence of Reel structures, each referencing a set of external Track Files, e.g. a sound or picture Track File, which are meant to be reproduced in parallel. Each Reel is analogous to a film reel and the Composition Playlist controls the order and timing of the playout of the Reels.

2 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 standard are encouraged to investigate the possibility of applying the most recent edition of the standards indicated below.

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

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

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

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

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

Internet Engineering Task Force (IETF) RFC2046 (November 1996) Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types.

Internet Engineering Task Force (IETF) (1996, November). RFC 2396 – Uniform Resource Identifiers (URI): Generic Syntax.

Internet Engineering Task Force (IETF) (2001, September). RFC 3174 – US Secure Hash Algorithm 1 (SHA-1).

Internet Engineering Task Force (IETF) (1997, May) RFC 2141 – URN Syntax.

Internet Engineering Task Force (IETF) (2001, April) RFC 4051 – Additional XML Security Uniform Resource Identifiers (URIs).

Internet Engineering Task Force (IETF) (2005, July). RFC 4122 – A Universally Unique Identifier (UUID) URN Namespace.

3 Overview

A composition, depicted in Figure 1, is a self-contained representation of a single complete D-Cinema work, such as a motion picture, or a trailer, or an advertisement, etc. It tangibly consists of a Composition Playlist file and one or more track files which contain the actual essence. Specifications of track file formats are beyond the scope of this document.

A Composition Playlist (CPL) is a document which specifies the manner in which track files are rendered. A CPL represents a composition as an ordered sequence of *Reels*. Each Reel contains one or more Assets, which identify Track File segments to be reproduced in parallel. In other words, it specifies the assembly of track files both in parallel, e.g. sound with picture, and in sequence; e.g., Reel 2 after Reel 1. The Composition Playlist is typically created under editorial control in the mastering environment and is then included in the D-Cinema package distributed to exhibition.

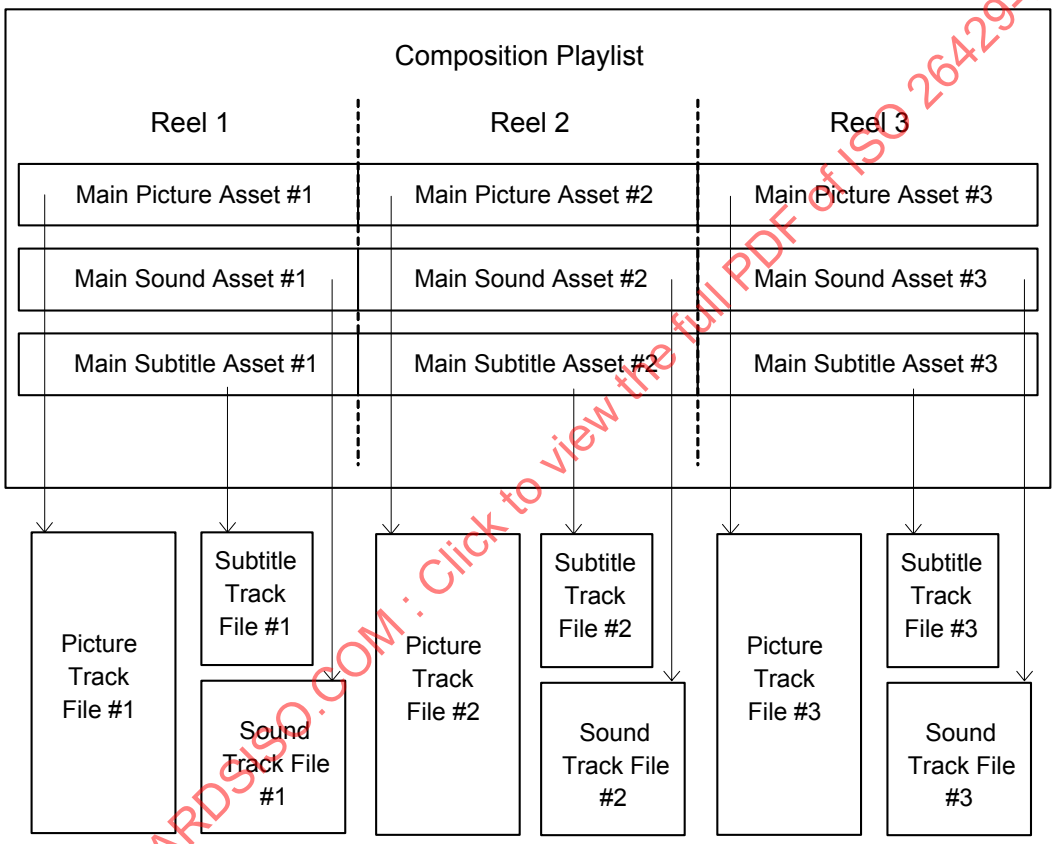

Figure 1 – Prototypical Composition Playlist

The structures defined in this document are represented using the Extensible Markup Language (XML) [XML 1.0], and specified using XML Schema [XML Schema Part 1: Structures] and [XML Schema Part 2: Datatypes]. This specification shall be associated with a unique XML namespace name [Namespaces in XML]. The namespace name shall be the string value "`http://www.smpte-ra.org/schemas/429-7/2006/CPL`". This namespace name conveys both structural and semantic version information, and serves the purpose of a traditional version number field.

Table 1 lists the XML namespace names used in this specification. Namespace names are represented as Uniform Resource Identifier (URI) values [RFC 2396]. These values shall be considered as simple strings, and applications should not attempt to resolve them as URLs.

Table 1 – XML Namespaces

Qualifier	URI
cpl	http://www.smpte-ra.org/schemas/429-7/2006/CPL
xs	http://www.w3.org/2001/XMLSchema
ds	http://www.w3.org/2000/09/xmldsig

The namespace qualifier values (*namespace prefixes* in XML jargon) used in this document (`cpl`, `xs`, `ds`), are not normative values. Implementations shall perform correctly with any XML compliant namespace prefix value that is associated with a URI from table 1.

Datatypes from other schemas that are used in this document will be prefixed with the appropriate namespace qualifier (e.g., `xs:dateTime`). See [XML Schema Part 2: Datatypes] and [XML-Signature Syntax and Processing] for further information about these types.

The MIME type [IETF RFC 2046] for a document containing a single Composition Playlist element as its root shall be "`text/xml`".

4 Terminology

The following terms are used to describe the features of this standard.

Clip: A contiguous set of Editable Units intended to be reproduced sequentially.

Composition: A complete artistic or informational motion picture work, such as a feature, or a trailer, or an advertisement, etc.

Editable Unit: The smallest temporal increment of access to Essence, e.g. a *frame* or a *sample*.

Edit Rate: A number of Editable Units to be reproduced during a temporal interval having a duration of exactly one (1.0) second. Because Edit Rate values are not always integer values and sometimes require many digits of precision, Edit Rate values are expressed as a rational number (the ratio of two integers).

Essence: The sound, picture and data resources that make up a Composition.

Native Duration: The total number of Editable Units in a Track File.

Native Start Point: The first Editable Unit of a Track File. All Track Files are viewed by a Composition Playlist as a sequence of Editable Units numbered from 0 (zero). Consequently, the Editable Unit number of the Native Start Point of a Track File will always be 0 (zero).

Native End Point: The last Editable Unit of a Track File.

Playable Region: The set of Editable Units within a Track File that are intended to be reproduced as part of a Composition. A Track File may contain Editable Units before and/or after the Playable Region.

Track File: A file containing a single Clip of simple Essence, such as sound, picture or subtitle essence.

Sample Rate: The number of essence samples per second. Sample Rate values are expressed as a rational number (the ratio of two integers).

Frame Rate: The number of frames per second. Frame Rate values are expressed as a rational number (the ratio of two integers).

Rational Number: A number value that is expressed as the ratio of two integers. This provides for the definition of precise values that are not subject to the inaccuracies of floating point representation.

5 Synchronization

The Composition Playlist defines an idealized playback timeline. As depicted in Figure 2, the timeline consists of a sequence of contiguous Reels. A Reel defines a temporal segment of the composition and consists of a set of single-essence Assets.

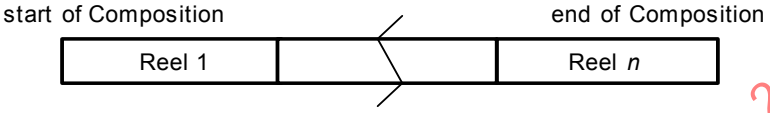

Figure 2 – Composition timeline

An Asset identifies a segment of a Track File to be reproduced in parallel with the other Assets in the same Reel. The Asset's Entry Point and Duration parameters define the sequence of Editable Units within the Track File that is to be reproduced (the Playable Region). For a given time offset T from the start of the Reel, the corresponding time offset T_n within a Track File F shall be equal to $(EntryPoint_F / EditRate_F) + T$. Editable Units across Track Files aligned with the same time offset T are synchronized and shall be reproduced simultaneously. Figure 3 illustrates the timing relationship between Assets within a Reel.

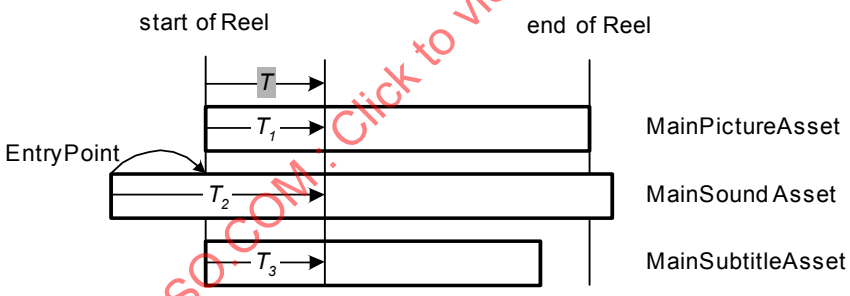

Figure 3 – Timing relationships within a Reel

At the start of a given Reel, playback of all Assets contained within the Reel shall start simultaneously at the Entry Point given for each respective Track File. The duration of a Reel shall be equal to the duration of the Reel's `MainPicture` Asset, or the Asset with the shortest duration if the `MainPicture` Asset is not present.

The Assets within each Reel shall be in editorial sync. In other words, the Composition Playlist timeline shall not compensate[1] for any processing delay that may occur in the exhibition environment, such as the internal image processing delay in a projector, and shall reflect the same time relationships the editor intended the audience to observe in exhibition.

[1] Any compensation for essence delay through the exhibition equipment is to be applied in-theatre, not in the mastering process. Such compensation may be applied by the digital cinema playback system, the cinema sound processor, or some other device.

6 CompositionPlaylist Structure

As depicted in Figure 4, a Composition Playlist shall be represented by a unique XML element, the `CompositionPlaylist` element. The Composition Playlist shall be encoded using the UTF-8 character encoding [XML 1.0].

The `CompositionPlaylist` element is defined using XML Schema in Section 10. The `CompositionPlaylist` element is illustrated in Figure 4 and the individual elements that comprise the `CompositionPlaylist` element are defined in the remaining sub-sections.

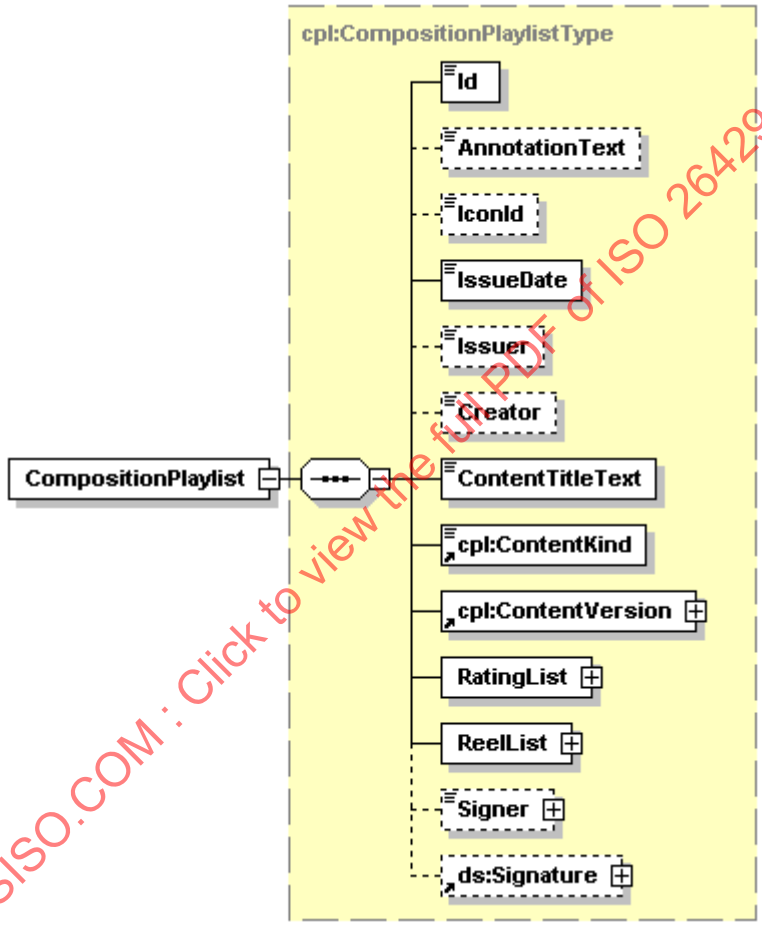

Figure 4 – CompositionPlaylist structure (Dotted lines denote optional elements)

6.1 Id

The Id element uniquely identifies the Composition Playlist for asset management purposes. It shall not uniquely identify the content represented by the composition – see Section 6.9. It shall be encoded as a urn:uuid per [RFC 4122].

6.2 AnnotationText [optional]

The `AnnotationText` element shall be a free-form, human-readable annotation describing the composition. It is meant strictly as a display hint to the user. The optional language attribute is an `xs:language` language

code and indicates the language used for the text. If the language attribute is not present, the default value `en` shall be used.

6.3 IconId [optional]

The `IconId` element uniquely identifies an external image resource containing a picture icon illustrating the composition. The icon may be rendered, for instance, from a frame of the underlying content. The `IconId` element shall be encoded as a `urn:uuid` per [RFC 4122]. The mapping of UUID values to actual image resources is beyond the scope of this document.

6.4 IssueDate

The `IssueDate` element shall be used to define the time and date at which the Composition Playlist was issued. It may be displayed to the user. It shall be encoded as an `xs:dateTime`.

6.5 Issuer [optional]

The `Issuer` element shall be a free-form, human-readable annotation that shall identify the entity that created the Composition Playlist. It is meant strictly for display to the user. The optional `language` attribute is an `xs:language` language code and indicates the text language of the content of the element. If the `language` attribute is not present, the default value `en` shall be used.

6.6 Creator [optional]

The `Creator` element shall be a free-form, human-readable annotation that shall identify the application used to create the Composition Playlist. It is meant strictly for display to the user. The optional `language` attribute is an `xs:language` language code and indicates the text language of the content of the element. If the `language` attribute is not present, the default value `en` shall be used.

6.7 ContentTitleText

The `ContentTitleText` 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. The optional `language` attribute is an `xs:language` language code and indicates the language of the content of the element. If the `language` attribute is not present, the default value `en` shall be used.

6.8 ContentKind

The `ContentKind` element defines the kind of material referred to by the Composition Playlist. It is meant to be both human and machine-readable. An optional `scope` attribute with default URI value `http://www.smpte-ra.org/schemas/429-7/2006/CPL#standard-content` determines the permissible values of the element. If the `scope` attribute is absent, or set to its default value, the content of the element shall match one of the values listed in Table 2; otherwise the content of the element is outside the scope of this specification but may be displayed to the user.

Table 2 – Content Kind

Kind	Description
feature	A theatrical feature.
trailer	Short (2 to 3 minutes) content promoting an upcoming theatrical feature.
test	Content used to test, calibrate or setup D-Cinema exhibition equipment.
teaser	Very short (typically less than 1 minute) content promoting an upcoming theatrical feature.
rating	Slate/still picture indicating the recommended age group permitted to view the content to follow. This rating is generally unique per country.
advertisement	Content promoting a product or service other than an upcoming feature.
short	Non advertising/promotional content (3 to 15 minutes) typically before a theatrical feature.
transitional	Extremely short content (1 to 15 seconds) separating unrelated compositions.
psa	Public service announcement.
policy	Content defining the code of conduct for patrons.

6.9 ContentVersion

The `ContentVersion` element defines the version of the content referred to by the composition, as opposed to the Composition Playlist `Id` element (Section 6.1) which uniquely identifies an instance of the Composition Playlist. Thus there may be two distinct compositions, with distinct `CompositionPlaylist Id` (Section 6.1) values, that refer to the same content and thus have the same `ContentVersion Id` value. This may occur, for example, if a composition is distributed to supercede a previous version. Similarly, while two compositions may share the same content title (Section 6.7), they may refer to two different versions, such as French (dubbed) and French (original), and therefore have two distinct `ContentVersion Id` values.

The `ContentVersion` element is meant to assist both users and software in scheduling and tracking content.

As shown in Figure 5, the `ContentVersion` element contains an `Id` element and a `LabelText` element, which are described in the following subsections.

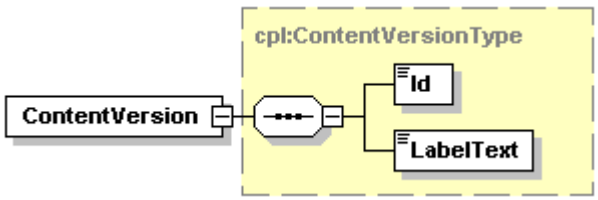

Figure 5 – ContentVersion structure (Dotted lines denote an optional element)

6.9.1 Id

The `Id` element shall identify the content contained in the Composition Playlist. It shall be a valid URN, per [RFC 2141].

6.9.2 LabelText

The `LabelText` element shall be a human readable label, e.g. “French (1.85 picture, 5.1 sound, dubbed)”, describing the content. The optional `language` attribute is an `xs:language` language code and indicates the text language of the content of the element. If the `language` attribute is not present, the default value `en` shall be used.

6.10 RatingList

The `RatingList` element shall contain an ordered list of zero or more `Rating` elements containing ratings associated with the composition.

Each `Rating` element, shown in Figure 6, contains an `Agency` and a `Label` element. Each element is meant to be both human and machine-readable. There shall be only one `Rating` element per given `Agency`.

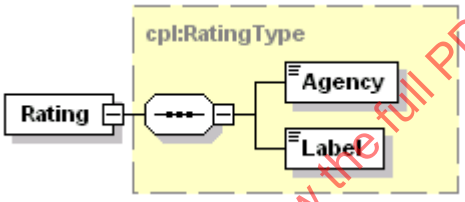

Figure 6 – Rating Element Structure (Dotted lines denote an optional element)

The `Agency` element shall contain a URI [RFC 2396] that uniquely identifies the agency issuing the rating. The `Label` element shall contain a textual representation of the rating, which may be displayed to the user. For each issuing agency, and hence unique URI, there are a number of permissible `Label` values. The specification of this mapping is beyond the scope of this document. However an informational sample mapping is shown in Table 3.

Table 3 – Example Ratings (Informative)

Agency	Labels
http://www.mpaa.org/2003-ratings	R, PG, PG-13, G, NC-17
http://rcq.qc.ca/2003-ratings	G, 13+, 16+, 18+

6.11 ReelList

The `ReelList` element shall contain an ordered list of `Reel` elements to be reproduced in sequence. The structure of the `Reel` element shall be as defined in Section 7.

6.12 Signer [optional]

The `Signer` element uniquely identifies the entity, and hence the public-private key pair, that digitally signed the Composition Playlist. It shall be an instance of the `KeyInfoType` type defined in [XML-Signature Syntax and Processing]. If the `Signer` element is present, then the `Signature` element shall also be present.

If X.509 certificates are used per [XML-Signature Syntax and Processing], then the `Signer` element shall contain one `X509Data` element containing one `X509IssuerSerial` element, which uniquely identifies the certificate used to sign the Composition Playlist. The Distinguished Name value in the X509IssuerName elements shall be compliant with RFC 2253 per [XML-Signature Syntax and Processing].

6.13 Signature [optional]

The `Signature` element shall contain a digital signature authenticating the Composition Playlist. If the `Signature` element is present, then the `Signer` element (6.12 above) shall also be present. The `Signature` element shall be an instance of the `ds:Signature` element defined in [XML-Signature Syntax and Processing]. The digital signature shall be *enveloped* and apply to the entire Composition Playlist. An enveloped signature is one that is attached to the document being signed. The signature is generated by the signer, as identified by the `Signer` element, using the signer's private key.

The standard `Signature` element is a highly flexible construct which can adapt to a wide range of applications. For the purpose of the Composition Playlist, it shall satisfy the following constraints:

- The `KeyInfo` element shall be present and shall contain the entire certificate chain for the signer.
- The `Object` element shall not be present and the URI attribute of the `Reference` element shall be set to: "" (empty string), as the signature is enveloped.
- The `Reference` element shall contain a single `DigestMethod` element, with its `Algorithm` attribute set to the URI value "`http://www.w3.org/2000/09/xmldsig#sha1`".
- The `Reference` element shall contain a single `Transform` element, with its `Algorithm` attribute set to the URI value "`http://www.w3.org/2000/09/xmldsig#enveloped-signature`".
- The `CanonicalizationMethod` shall be set to the URI value "`http://www.w3.org/TR/2001/REC-xml-c14n-20010315`".
- The `SignatureMethod` shall be set to the URI value "`http://www.w3.org/2001/04/xmldsig-more#rsa-sha256`" [RFC 4051].

Note that the URI values above shall be considered as simple strings and applications should not attempt to resolve them as URL values.

If X.509 certificates are used per [XML-Signature Syntax and Processing], then the entire certificate chain shall be carried in the `KeyInfo` element as a sequence of `X509Data` elements. Each of the `X509Data` elements shall correspond to one certificate in the chain, and contain one `X509IssuerSerial` element and one `X509Certificate` element. The Distinguished Name value in all X509IssuerName elements shall be compliant with RFC 2253 per [XML-Signature Syntax and Processing].

7 Reel Structure

Each Reel shall consist of a number of Assets, each corresponding to a particular aspect of the D-Cinema presentation. While this specification defines a number of asset types, additional asset types may be added in the future – see Sections 7.3 and 9.2. The `Reel` element is defined using XML Schema in Section 10. The `Reel` element is illustrated in Figure 7 and the individual elements that comprise the `Reel` element are defined in the remaining sub-sections.

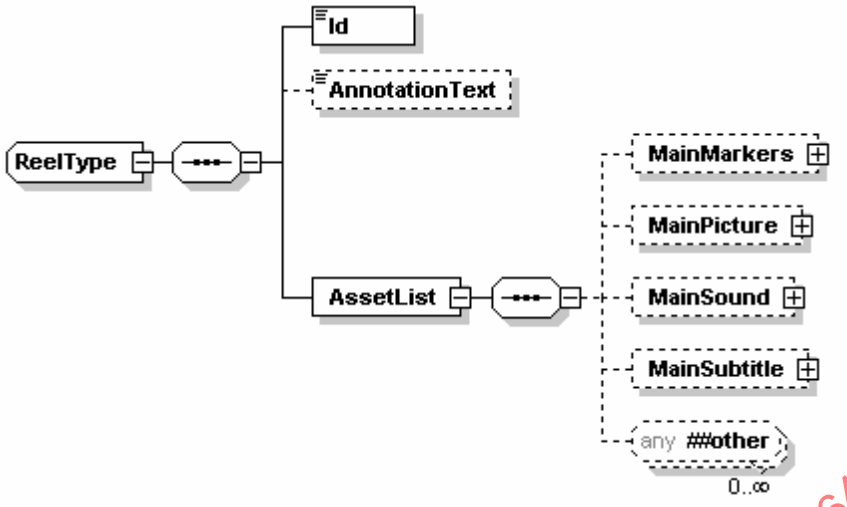

Figure 7 – Reel Structure (Dotted lines denote an optional element)

7.1 Id

The Id element uniquely identifies the reel for asset management purposes. It shall be encoded as a `urn:uuid` per [RFC 4122].

7.2 AnnotationText [optional]

The `AnnotationText` element shall be a free-form, human-readable, text annotation associated with the Reel. It is meant strictly as a display hint to the user. The optional `language` attribute is an `xs:language` language code and indicates the text language of the content of the element. If the `language` attribute is not present, the default value `en` shall be used.

7.3 AssetList

The `AssetList` element shall contain the ordered list of media assets that will be reproduced in parallel during reel playback. Each asset shall be uniquely identified by a UUID per [RFC 4122]. The defining specification for each Track File shall specify the location of the identifying UUID.

Each child element of the `AssetList` element shall be derived from the datatype `GenericAssetType` defined in Section 8.1. A number of standard assets are defined in the subsequent sections. As additional assets are defined; e.g., captions, the `AssetList` element may be extended by introducing new elements. These elements shall be associated with a namespace name distinct from that associated with this document and may be ignored by implementations compliant with this specification (See Section 7.3.5 below).

7.3.1 MainMarkers [optional]

The `MainMarkers` element defines markers, e.g. `FFOC`, `LFOC`..., associated with the main portion of the theatrical presentation, i.e., `MainPicture` and `MainSound` assets. Markers shall be referenced from the start of the reel they are associated with. The `MainMarkers` element shall be an instance of `MarkersAssetType` and its structure is described in more detail in Section 8.3.

7.3.2 MainPicture [optional]

The `MainPicture` element defines the picture essence to be projected onto the main screen. The actual picture essence is contained in an external Track File. The `MainPicture` element shall be an instance of `PictureTrackFileAssetType` and its structure is defined in Section 8.4.

7.3.3 MainSound [optional]

The `MainSound` element defines the sound essence to be reproduced in the auditorium. The actual sound essence is contained in an external Track File. The `MainSound` element shall be an instance of `SoundTrackFileAssetType` and its structure is defined in Section 8.5.

7.3.4 MainSubtitle [optional]

The `MainSubtitle` element defines the Subtitle essence to be reproduced on the main screen in the auditorium. The actual Subtitle essence is contained in an external Track File. The `MainSubtitle` element shall be an instance of `SubtitleTrackFileAssetType` and its structure is defined in Section 8.6.

7.3.5 Extensions (New Asset Types)

Extension elements shall be used to represent asset types not defined in this document. Zero or more extension elements may be present in the `AssetList`. When present, extension elements shall be located after any elements defined by this document. When present, extension elements shall have names that belong to a namespace different than the namespace declared by this document. Implementations may ignore extension elements belonging to an unknown namespace.

Extension elements shall directly or indirectly extend `GenericAssetType` (see Section 8.1). Extension elements may extend any type defined in this document which is derived from `GenericAssetType`.

INFORMATIVE NOTE – Extension elements should have unique, descriptive names and should appear only once in a given Reel. Extension specifications that allow multiple instances of an element in a Reel should provide both a means of differentiating instances within a Reel and a means of linking related instances in separate Reels.

8 Asset Structure

Reel Asset elements share common attributes, such as `Duration`, and are therefore specified as a set of types derived from a common structure, namely the `GenericAssetType` structure (Section 8.1). A number of Asset elements also reference external files, in which case they are derived from the `TrackFileAssetType` structure (Section 8.2). The Asset datatypes are defined using XML Schema in Section 10.

8.1 GenericAssetType

The `GenericAssetType` describes a generic asset intended to be reproduced as part of a Reel. It is illustrated in Figure 8. Individual child elements are defined in the following subsections.

8.1.1 Id

The `Id` element uniquely identifies the Asset. It shall be encoded as a `urn:uuid` per [RFC 4122]. If the Asset refers to an external resource, such as a Track File, the UUID value shall identify the resource.

Mapping of UUID values to actual resources is beyond the scope of this document.

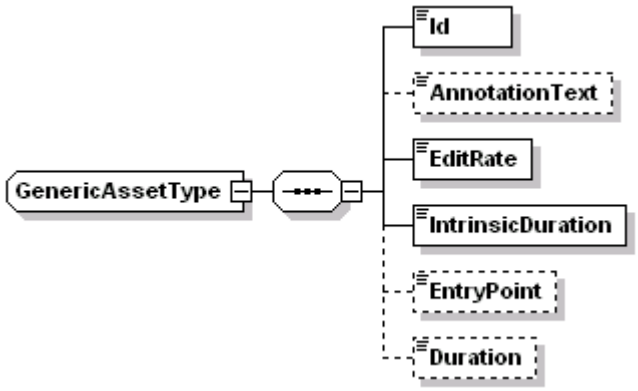

Figure 8 – GenericAssetType Structure (Dotted lines denote an optional element)

8.1.2 AnnotationText [optional]

The `AnnotationText` element shall be a free-form, human-readable text annotation associated with the asset. It meant strictly as a display hint to the user. The optional `language` attribute is an `xs:language` language code and indicates the text language of the content of the element. If the language attribute is not present, the default value `en` shall be used.

8.1.3 EditRate

The `EditRate` element defines the Edit Rate of the Asset. It shall be in units of inverse seconds and represented as a Rational Number. The `IntrinsicDuration`, `EntryPoint` and `Duration` parameters shall be expressed in units of 1/EditRate (i.e. as integer values). If the Asset refers to an external resource, `EditRate` may differ from the actual Edit Rate or Sample Rate of the underlying essence.

8.1.4 IntrinsicDuration

The `IntrinsicDuration` element shall define the Native Duration of the Asset, as illustrated in Figure 9. It shall not take into account the values of the `EntryPoint` and `Duration` elements. Unless the optional `EntryPoint` and `Duration` parameters are specified, playback of the asset shall start at the Native Start Point and terminate at the Native End Point of the Track File. `IntrinsicDuration` shall be expressed in units of 1/EditRate; i.e., as a count of Editable Units.

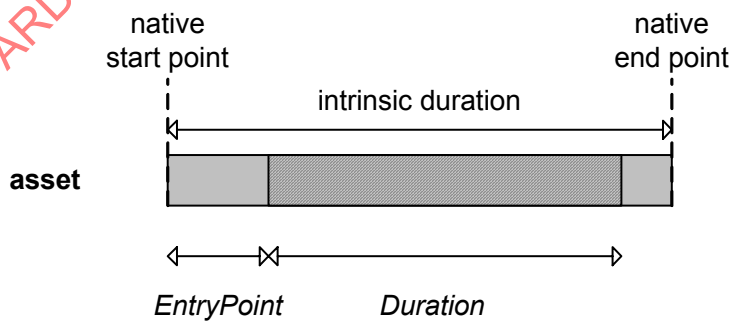

Figure 9 – Asset Timing Parameters

8.1.5 EntryPoint [optional]

This element shall only be present when the Asset refers to an external resource such as a Track File.

The `EntryPoint` element identifies the Edit Unit where playback shall start (the first editable unit of the Playable Region). It shall be encoded as an integer number and shall be expressed in units of 1/EditRate; i.e., as a count of Editable Units. This element shall be required if the desired Entry Point is greater than 0 (zero). If this element is not present, a value of 0 shall be assumed and Asset playback shall start at the Native Start Point of the resource.

8.1.6 Duration [optional]

This element shall only be present when the Asset refers to an external resource such as a Track File.

The `Duration` element defines the duration of the Playable Region of the asset. It shall be encoded as an integer number and shall be expressed as an integer number in units of 1/EditRate; i.e., as a count of Editable Units. If present, this value shall be an integer between 0 (zero) and `IntrinsicDuration` – `EntryPoint` (the number of edit units between the `EntryPoint` and the Native End Point the Track File). If this element is not present, Asset playback shall stop after (`IntrinsicDuration` – `EntryPoint`)/`EditRate` seconds; i.e., at the Native End Point of the Asset.

8.2 TrackFileAssetType

The `TrackFileAssetType` element, as illustrated in Figure 10, shall be derived from `GenericAssetType`. It describes an asset based on an external file such as a Picture or Sound Track File.

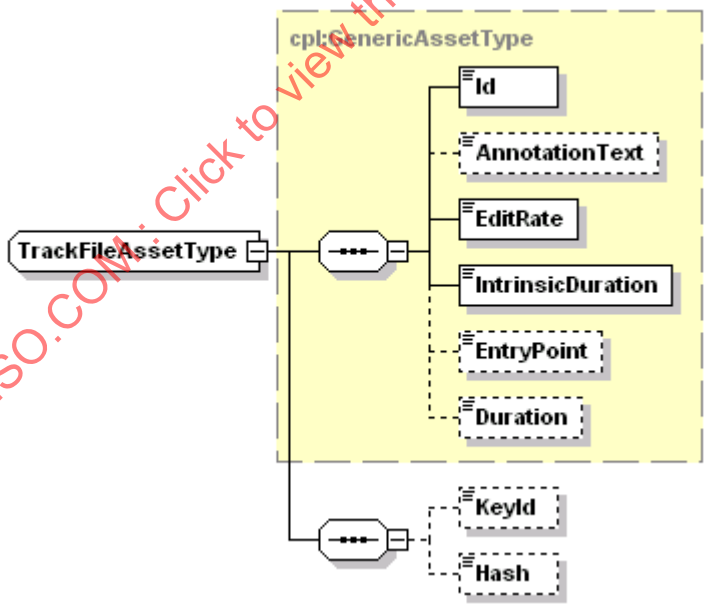

Figure 10 – Track File Asset Structure (Dotted lines denote an optional element)

8.2.1 KeyId [optional]

The `KeyId` element uniquely identifies the cryptographic key used to encrypt the underlying track file. This element shall contain a key identifier encoded as a `urn:uuid` value. `KeyId` 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.

8.2.2 Hash [optional]

The `Hash` element shall contain the hash (message digest) of the underlying track file computed using the SHA-1 message digest algorithm [RFC 3174]. When authenticated by the digital signature in the Composition Playlist (see 6.13), it may be used to verify the integrity and authenticity of the underlying track file. The resulting 160-bit integer shall be encoded using Base64 representation [RFC 2045].

8.3 MarkerAssetType

The `MarkerAssetType` element, as illustrated in Figure 11, shall be derived from the `GenericAssetType`. It describes the content markers; e.g., `FFOC`, associated with a reel. An instance of the `MarkerAssetType` is the `MainMarkers` element.

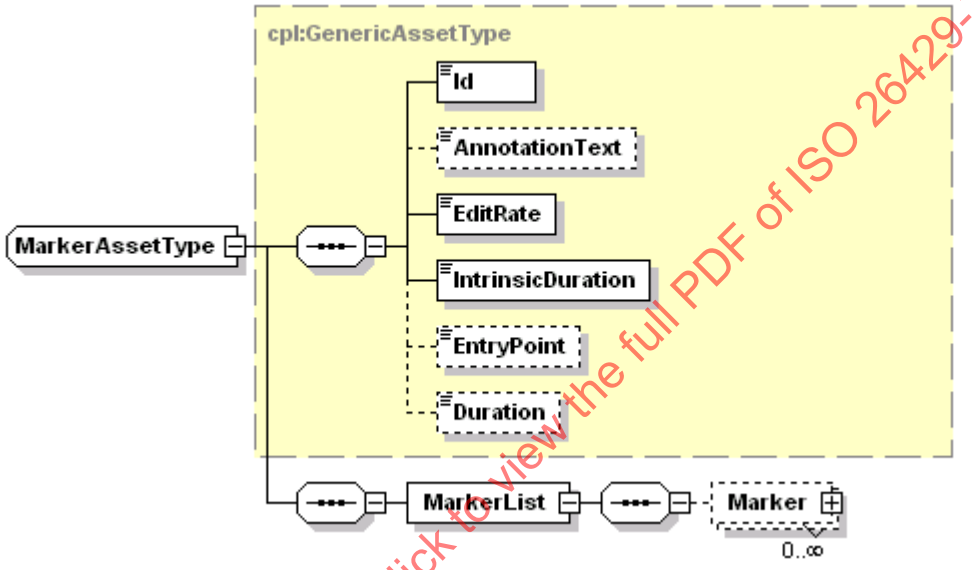

Figure 11 – Content Marker Asset Structure (Dotted lines denote an optional element)

Just as for picture and sound assets, the marker asset has a timeline. The Offset of each Marker (see 8.3.1.3) is the position from the start of the timeline and the Intrinsic Duration of the timeline shall correspond to the Offset of the last Marker.

8.3.1 MarkerList

The `MarkerList` element shall contain a list of `Marker` elements. The structure of each individual marker is illustrated in Figure 12. The members of the `Marker` element are defined in the following subsections.

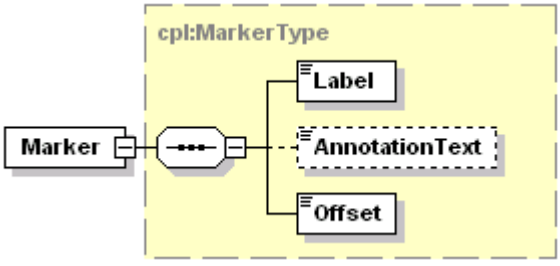

Figure12 – Marker Structure (Dotted lines denote an optional element)

8.3.1.1 Label

The `Label` element shall contain a textual representation of the marker. An optional scope attribute with default URI value of "`http://www.smpte-ra.org/schemas/429-7/2006/CPL#standard-markers`" determines the permissible values of the element. If the attribute is absent or set to its default value, the content of the element shall match one of the values listed in Table 4; otherwise the content of the element is outside the scope of this specification but may be displayed to the user. Note that the string above shall be considered a simple string and applications should not attempt to resolve it as a URL value.

Table 4 – Standard Marker Labels

Marker	Description
`FFOC`	First Frame of Composition. The first frame of a composition that is intended for display.
`LFOC`	Last Frame of Composition. The last frame of a composition that is intended for display.
`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.
`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.
`FFOI`	First Frame of Intermission.
`LFOI`	Last Frame of Intermission.
`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.
`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.
`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.
`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.
`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.
`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.

8.3.1.2 AnnotationText [optional]

The `AnnotationText` element shall be a free-form, human-readable annotation associated with the marker. It meant strictly as a display hint to the user. The optional `language` attribute is a standard XML language code and indicates the text language of the content of the element – see Section 10. If the `language` attribute is not present, the default value `en` shall be used.

8.3.1.3 Offset

The `Offset` element defines the absolute position of the marker from the start of the marker asset. It shall be represented as integer number of 1/EditRate units, as inherited from the `GenericAssetType`.

8.4 PictureTrackFileAssetType

The `PictureTrackFileAssetType`, as illustrated in Figure 13, shall be derived from the `TrackFileAssetType`. It describes a Track File containing picture essence. An instance of the `PictureTrackFileAssetType` is the `MainPicture` element.

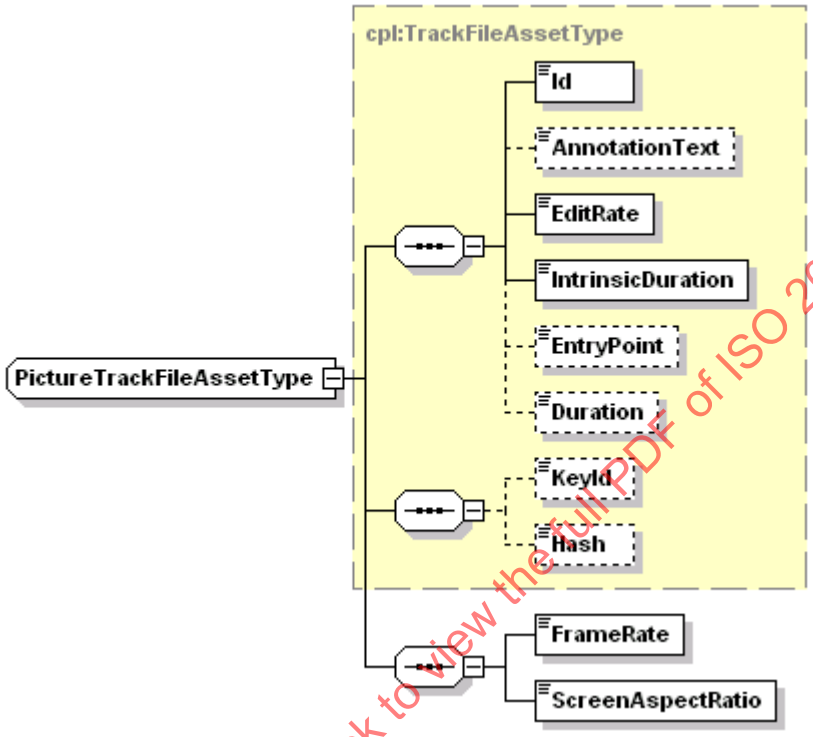

Figure 13 – Picture Track File Asset Structure (Dotted lines denote an optional element)

The elements defined below replicate values contained in the underlying track file and shall remain consistent with the content of the underlying track file at all times. They are included in the Composition Playlist to alleviate the need for theater management software to access and parse individual track files in order to display the values to users. In the event an inconsistency exists, the values contained in the underlying track file shall take precedence.

8.4.1 FrameRate

The `FrameRate` element shall contain the frame rate of the underlying picture track file. As described above, it is included in the Composition Playlist for convenience only. It shall be encoded as a rational number of frames per second.

8.4.2 ScreenAspectRatio

The `ScreenAspectRatio` element shall define the aspect ratio of the picture information contained in the underlying picture track file. As described above, it is included in the Composition Playlist for convenience. It is represented as a rational number and applications may convert the ratio to a decimal number to match current film practice.

8.5 SoundTrackFileAssetType

The `SoundTrackFileAssetType`, as illustated in Figure 14, is derived from `TrackFileAssetType`. It describes a Track File containing sound essence. An instance of the `SoundTrackFileAssetType` is the `MainSound` element.

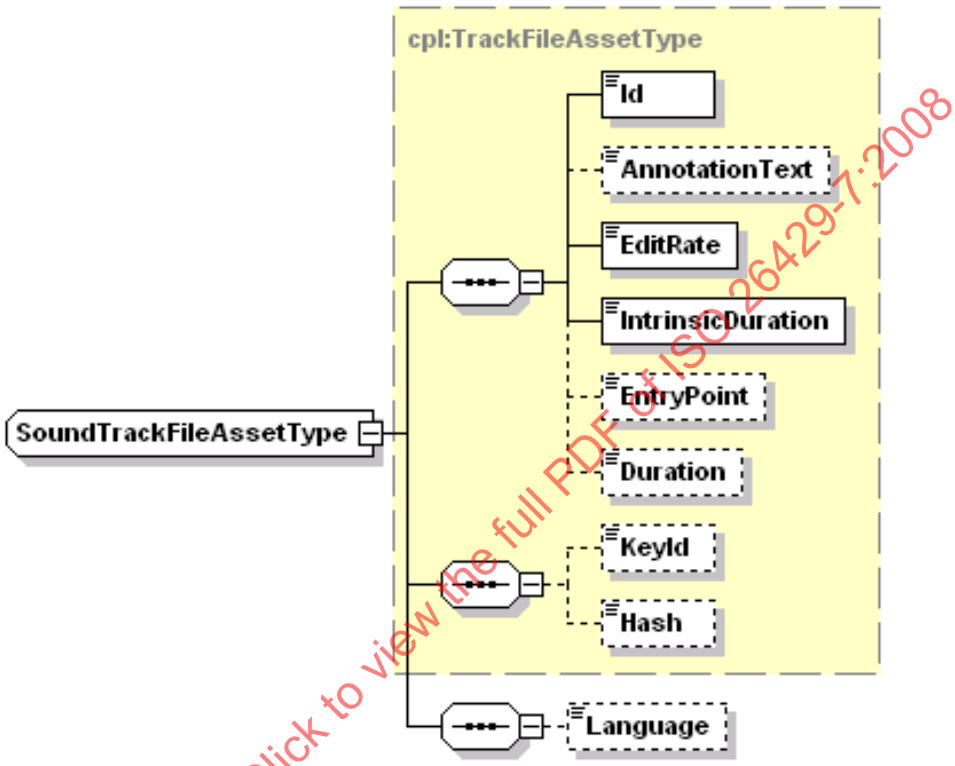

Figure 14 – Sound Track File Asset structure (Dotted lines denote an optional element)

The elements defined below replicate values contained in the underlying track file and shall remain consistent with the content of the underlying track file at all times. They are included in the Composition Playlist to alleviate the need for theater management software to access and parse individual track files when scheduling content. In the event an inconsistency exists, the values contained in the underlying track file shall take precedence.

8.5.1 Language [optional]

The `Language` element shall reflect the primary spoken language of the sound material of the underlying sound track file. The element value is encoded as an `xs:language` language code and indicates the spoken language of the content. The absence of the element shall indicate that no spoken language is associated with the asset.

8.6 SubtitleTrackFileAssetType

The `SubtitleTrackFileAssetType` element, as illustrated in Figure 15, is derived from `TrackFileAssetType`. It describes the subtitle material associated with the reel. An instance of the `SubtitleTrackFileAssetType` is the `MainSubtitle` element.

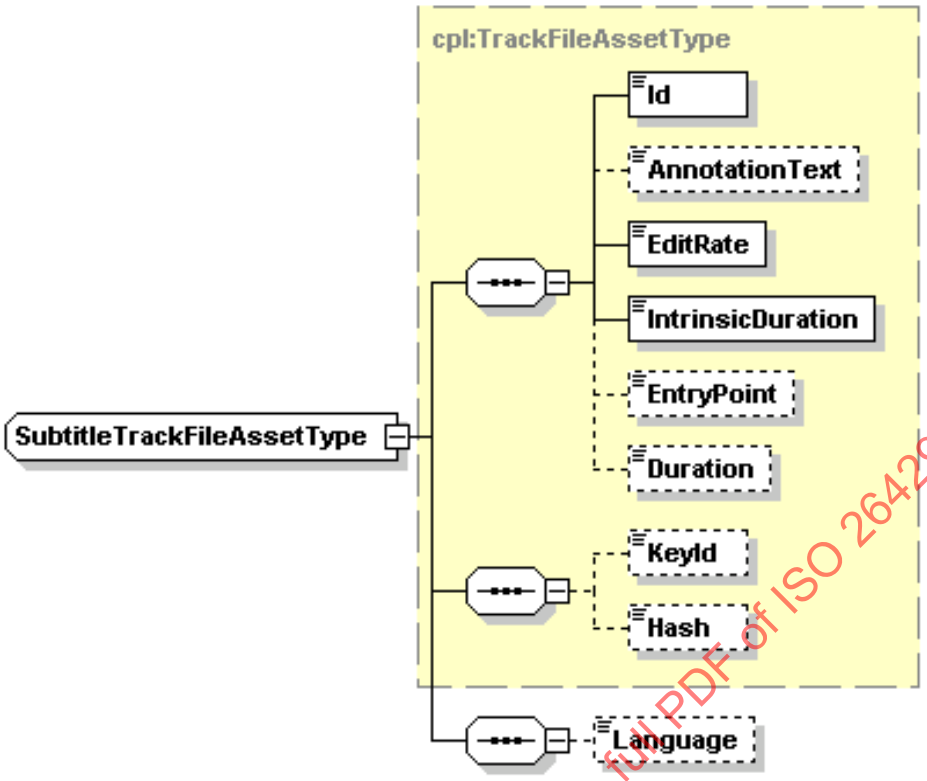

Figure 15 – Subtitle Track File Asset structure (Dotted lines denote an optional element)

The elements defined below replicate values contained in the underlying track file and shall remain consistent with the content of the underlying track file at all times. They are included in the Composition Playlist to alleviate the need for theater management software to access and parse individual track files when scheduling content. In the event an inconsistency exists, the values contained in the underlying track file shall take precedence.

8.6.1 Language [optional]

The `Language` element shall reflect the primary text language used by the Subtitle essence. The absence of the element shall indicate that no primary text language is associated with the asset. It is represented as an `xs:language` value.

9 CPL Constraints

9.1 Content Markers

The Content Markers of Section 8.3 are intimately related to the content they describe and hence most efficiently created at the time of mastering. In fact, a human operator would have to manually scrub through the content upon reception to create the same markers in an exhibition environment.

To reduce burden on exhibition operators and devices, Table 5 lists the content markers that should be inserted at mastering, as a function of the kind of content described by the Composition Playlist. If the underlying content does not support a particular marker, e.g. a particular feature does not have title credits, then the corresponding content marker shall not be created, even if present in Table 5. No more than one instance of each marker shall be present in any given Composition Playlist.

Table 5 – Content Markers

Content Kind	Markers
feature	FFOC, LFOC, FFTC, LFTC, FFOI, LFOI, FFEC, FFOB, LFOB, LFEC
trailer	FFOC, LFOC
test	FFOC, LFOC
teaser	FFOC, LFOC
rating	FFOC, LFOC, FFOB, LFOB
advertisement	FFOC, LFOC
short	FFOC, LFOC
transitional	FFOC, LFOC
psa	FFOC, LFOC
policy	FFOC, LFOC

9.2 Minimum Reel Duration

The duration of any asset contained in a reel, as indicated by the `Duration` and `IntrinsicDuration` elements, shall be no less than one second.

10 XML Schema

The XML Schema document presented in this section normatively defines the structure of a Composition Playlist using a machine-readable language. While this schema is intended to faithfully represent the structure presented in the normative prose portions (Sections 3 through 9) of this document, conflicts in definition may occur. In the event of such a conflict, the normative prose shall be the authoritative expression of the standard.

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema targetNamespace="http://www.smpte-ra.org/schemas/429-7/2006/CPL"
              xmlns:cpl="http://www.smpte-ra.org/schemas/429-7/2006/CPL"
              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.w3.org/2000/09/xmldsig#"
                           schemaLocation="http://www.w3.org/TR/2002/REC-xmldsig-core-
                           20020212/xmldsig-core-schema.xsd"/>
  <xs:import namespace="http://www.w3.org/XML/1998/namespace"
                           schemaLocation="http://www.w3.org/2001/03/xml.xsd"/>

 <!-- CompositionPlaylist -->
  <xs:element name="CompositionPlaylist" type="cpl:CompositionPlaylistType"/>
  <xs:complexType name="CompositionPlaylistType">
    <xs:sequence>
      <xs:element name="Id" type="cpl:UUID"/>
      <xs:element name="AnnotationText" type="cpl:UserText" minOccurs="0"/>
      <xs:element name="IconId" type="cpl:UUID" minOccurs="0"/>
      <xs:element name="IssueDate" type="xs:dateTime"/>
      <xs:element name="Issuer" type="cpl:UserText" minOccurs="0"/>
      <xs:element name="Creator" type="cpl:UserText" minOccurs="0"/>
      <xs:element name="ContentTitleText" type="cpl:UserText"/>
      <xs:element name="ContentKind" type="cpl:ContentKindType"/>
      <xs:element name="ContentVersion" type="cpl:ContentVersionType"/>
      <xs:element name="RatingList">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="Rating" type="cpl:RatingType" minOccurs="0"
                        maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="ReelList">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="Reel" type="cpl:ReelType" 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>
```

```
<!-- Marker -->
<xs:complexType name="MarkerType">
   <xs:sequence>
      <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/429-7/2006/CPL#standard-
                           markers"/>
               </xs:extension>
            </xs:simpleContent>
         </xs:complexType>
      </xs:element>
      <xs:element name="AnnotationText" type="cpl:UserText" minOccurs="0"/>
      <xs:element name="Offset">
         <xs:simpleType>
            <xs:restriction base="xs:long">
               <xs:minInclusive value="0"/>
            </xs:restriction>
         </xs:simpleType>
      </xs:element>
   </xs:sequence>
</xs:complexType>

<!-- Rating -->
<xs:complexType name="RatingType">
   <xs:sequence>
      <xs:element name="Agency" type="xs:anyURI"/>
      <xs:element name="Label" type="xs:string"/>
   </xs:sequence>
</xs:complexType>

<!-- ContentKind -->
<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/429-7/2006/CPL#standard-content"/>
      </xs:extension>
   </xs:simpleContent>
</xs:complexType>

<!-- ContentVersion -->
<xs:complexType name="ContentVersionType">
   <xs:sequence>
      <xs:element name="Id" type="xs:anyURI"/>
      <xs:element name="LabelText" type="cpl:UserText"/>
   </xs:sequence>
</xs:complexType>

<!-- Reel -->
<xs:complexType name="ReelType">
   <xs:sequence>
      <xs:element name="Id" type="cpl:UUID"/>
      <xs:element name="AnnotationText" type="cpl:UserText" minOccurs="0"/>
      <xs:element name="AssetList">
         <xs:complexType>
            <xs:sequence>
               <xs:element name="MainMarkers" type="cpl:MarkerAssetType" minOccurs="0"/>
               <xs:element name="MainPicture" type="cpl:PictureTrackFileAssetType"
                           minOccurs="0"/>
               <xs:element name="MainSound" type="cpl:SoundTrackFileAssetType" minOccurs="0"/>
               <xs:element name="MainSubtitle" type="cpl:SubtitleTrackFileAssetType"
                           minOccurs="0"/>
               <xs:any namespace="##other" minOccurs="0" maxOccurs="unbounded"
                           processContents="lax" />
            </xs:sequence>
         </xs:complexType>
      </xs:element
   </xs:sequence>
```

```
</xs:complexType>


<!-- GenericAssetType -->
<xs:complexType name="GenericAssetType" abstract="1">
   <xs:sequence>
      <xs:element name="Id" type="cpl:UUID"/>
      <xs:element name="AnnotationText" type="cpl:UserText" minOccurs="0"/>
      <xs:element name="EditRate" type="cpl:Rational"/>
      <xs:element name="IntrinsicDuration" type="xs:long"/>
      <xs:element name="EntryPoint" type="xs:long" minOccurs="0"/>
      <xs:element name="Duration" type="xs:long" minOccurs="0"/>
   </xs:sequence>
</xs:complexType>

<!-- TrackFileAssetType -->
<xs:complexType name="TrackFileAssetType" abstract="1">
   <xs:complexContent>
      <xs:extension base="cpl:GenericAssetType">
         <xs:sequence>
            <xs:element name="KeyId" type="cpl:UUID" minOccurs="0"/>
            <xs:element name="Hash" type="xs:base64Binary" minOccurs="0"/>
         </xs:sequence>
      </xs:extension>
   </xs:complexContent>
</xs:complexType>

<!-- SountTrackFileAssetType -->
<xs:complexType name="SoundTrackFileAssetType">
   <xs:complexContent>
      <xs:extension base="cpl:TrackFileAssetType">
         <xs:sequence>
            <xs:element name="Language" type="xs:language" minOccurs="0"/>
         </xs:sequence>
      </xs:extension>
   </xs:complexContent>
</xs:complexType>

<!-- SubtitleTrackFileAssetType -->
<xs:complexType name="SubtitleTrackFileAssetType">
   <xs:complexContent>
      <xs:extension base="cpl:TrackFileAssetType">
         <xs:sequence>
            <xs:element name="Language" type="xs:language" minOccurs="0"/>
         </xs:sequence>
      </xs:extension>
   </xs:complexContent>
</xs:complexType>

<!-- PictureTrackFileAssetType -->
<xs:complexType name="PictureTrackFileAssetType">
   <xs:complexContent>
      <xs:extension base="cpl:TrackFileAssetType">
         <xs:sequence>
            <xs:element name="FrameRate" type="cpl:Rational"/>
            <xs:element name="ScreenAspectRatio" type="cpl:Rational"/>
         </xs:sequence>
      </xs:extension>
   </xs:complexContent>
</xs:complexType>
```