SMPTE STANDARD

D-Cinema Packaging — Packing List

Table of Contents

Approved
June 25, 2007

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.

SMPTE 429-8 was prepared by Technology Committee DC28.

1 Scope

This standard specifies the data format for interchange of a Packing List for Digital Cinema applications.

The electronic or physical form of a complete package described by a Packing List is beyond the scope of this standard.

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.

3 Normative References

The following standards contain provisions which, through reference in this text, constitute provisions of this recommended practice. 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. World Wide Web Consortium (W3C) (2004, February 4). *Extensible Markup Language (XML) 1.0 (Third Edition).*

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

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

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

5. Internet Engineering Task Force (IETF) RFC3174 (September 2001) *US Secure Hash Algorithm 1*

6. Internet Engineering Task Force (IETF) RFC2045 (November 1996) *Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies*

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

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

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

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

4 Overview

The packing list specifies the contents of a distribution package. A distribution package shall contain one packing list together with Composition Playlist assets, essence assets and other assets as needed to complete the package. The packing list has a list of elements that define the distribution package together with a list of references to all the assets in the package. This list contains the Ids that uniquely identify each asset in the package.

Figure 1 illustrates the abstract form of a complete package for a trailer and a single reel feature.

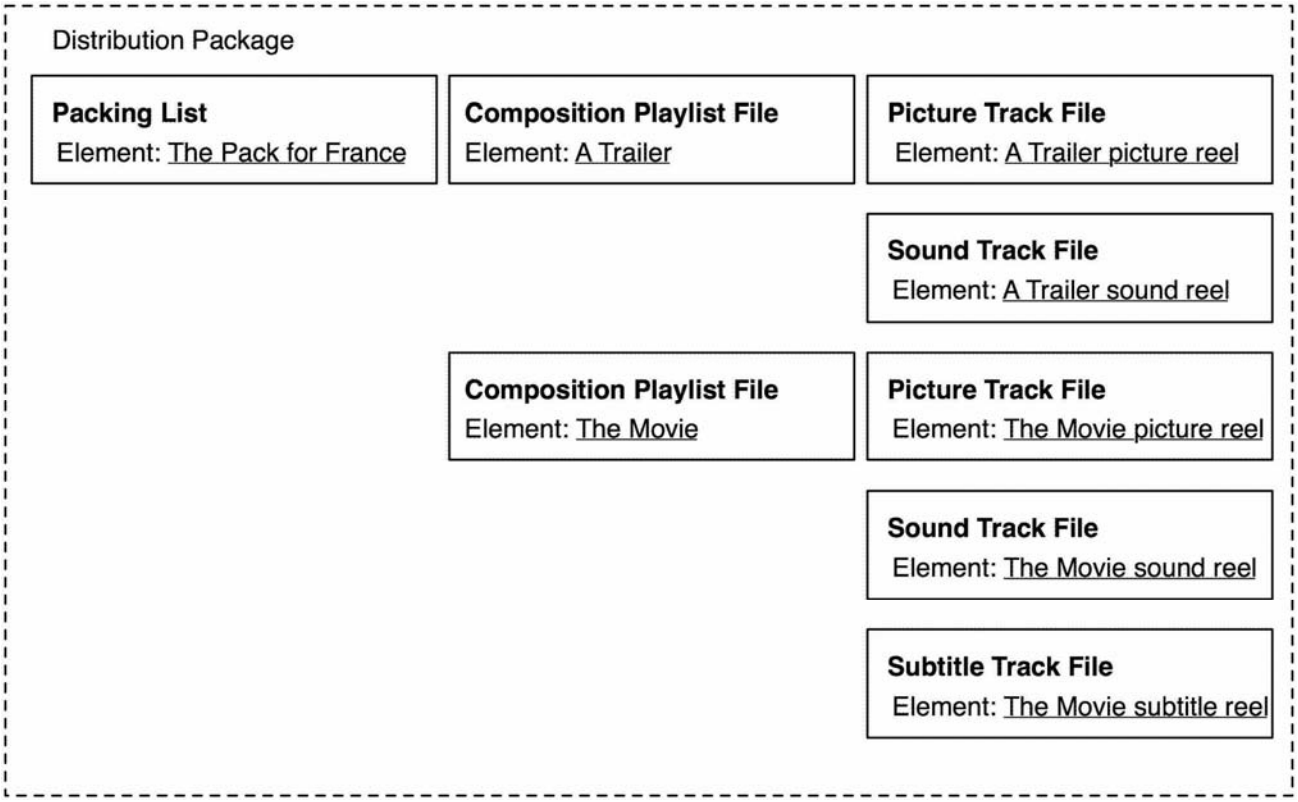


Figure 1 – Typical Distribution Package (Informative)

Figure 2 illustrates the abstract form of a partial Packing List, in which a replacement “reel” is being sent to theatres. This may be created, for example, to distribute a change in the end credits of a feature. Note that this partial package includes picture, sound and subtitle track files, and a Composition Playlist file that references those files as well as files from a previously delivered package. The example package shown in Figure 2 is not related to the example package shown in Figure 1.

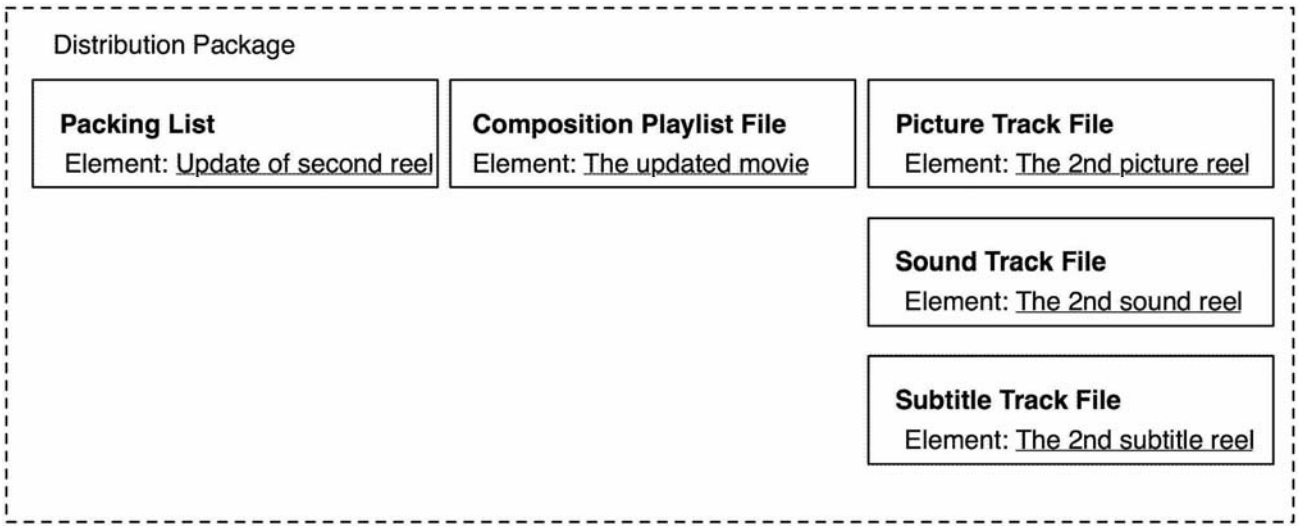


Figure 2 – Typical Partial Distribution Package (Informative)

4.1 Use of XML Language

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 Datatypes [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/429-8/2007/PKL". 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][1].

Table 1 – XML Namespaces

Qualifier	*URI*
pkl	http://www.smpte-ra.org/schemas/429-8/2007/PKL
xs	http://www.w3.org/2001/XMLSchema
ds	http://www.w3.org/2000/09/xmldsig#

The URIs found in Table 1 are normative. The namespace qualifier values (also called namespace prefixes in XML jargon) used in Table 1 and elsewhere in this document, namely "pkl", "xs" and "ds", are not normative. Specifically, they may be replaced in instance documents by any XML compliant namespace prefix. In other words, implementations shall expect any arbitrary XML compliant namespace prefix value that is associated with a URI from table 1.

[1] Readers unfamiliar with URI values as XML namespace names should be aware that although a URI value begins with a "method" element ("http" in this case), the value is designed primarily to be a unique string and does not necessarily correspond to an actual on-line resource. Applications implementing this standard should not attempt to resolve URI values on-line.

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 `PackingList` element as its root shall be "`text/xml`".

5 PackingList Structure

A Packing List shall be encoded as an XML document [XML 1.0]. The top-level element shall be designated PackingList, and is described in Figure 3.

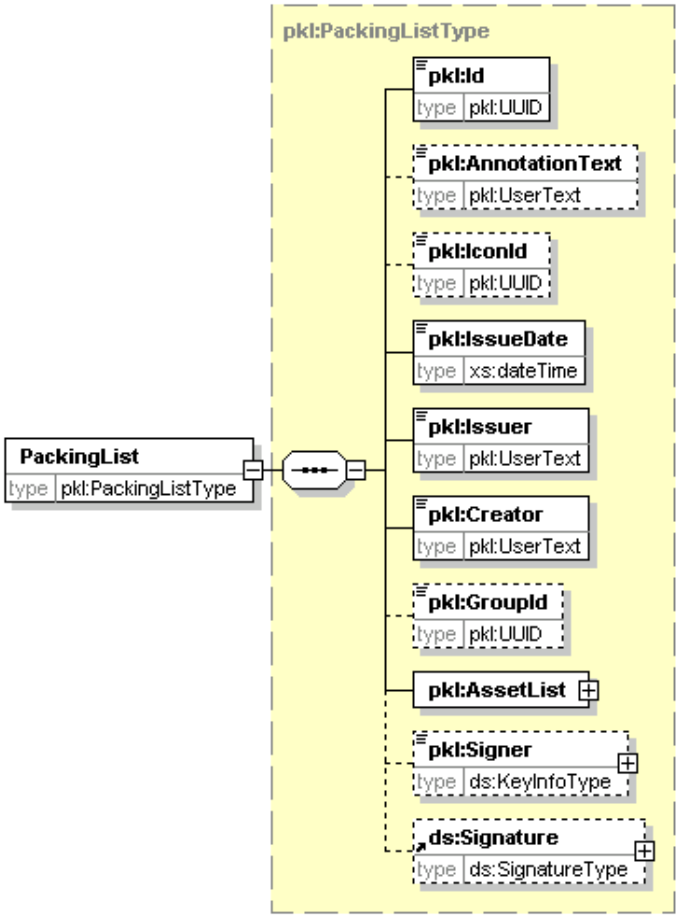


Figure 3 – Packing List structure. Dotted lines denote an optional element.

5.1 Id

The `Id` element uniquely identifies the packing list for asset management purposes. Each unique Packing List shall have a distinct Id. This will allow easy differentiation between Packing Lists. The `Id` shall be encoded as a urn:UUID [RFC 4122].

5.2 AnnotationText [optional]

The `AnnotationText` element shall be a free-form, human-readable annotation describing the distribution package. It is meant strictly as a displayed guidance for 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.

5.3 IconId [optional]

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

5.4 IssueDate

The `IssueDate` element indicates the time and date at which the Packing List was issued. The `IssueDate` shall be encoded as an `xs:dateTime` value.

5.5 Issuer

The `Issuer` element shall be a free-form, human-readable annotation describing the person or company that created the Packing List. It is meant strictly as a displayed guidance for 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.

5.6 Creator

The `Creator` element shall be a free-form, human-readable annotation describing the person, facility or system (hardware/software) that created the Packing List. It is meant strictly as a displayed guidance for 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.

5.7 GroupId [optional]

The `GroupId` element is used to create associations between packages. When present, the element shall contain a urn:UUID value [RFC 4122]. The presence of the element shall indicate to a receiver that the package may be associated for asset management purposes with any other package containing the same `GroupId` value. The exact meaning of the association is beyond the scope of this document, but in general a packing list with no `GroupId` element should contain a complete set of assets (i.e., there are no unresolved references between assets). Two or more packing lists with matching GroupId elements should contain assets that are related (e.g., referentially).

5.8 AssetList

The AssetList element contains the list of Asset elements contained in the package. The structure of the Asset element is described in Section 6 of this document. The order of Asset elements in the list shall not be significant.

5.9 Signer [optional]

The Signer element uniquely identifies the entity, and hence the public-private key pair that digitally signed the Packing List. 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 Packing List.

5.10 Signature [optional]

The Signature element shall contain a digital signature authenticating the Packing List. If the Signature element is present, then the Signer element (see 5.9, above) shall also be present. The Signature element shall be an instance of the ds:Signature element defined in the [XML Signature Syntax and Processing]. The digital signature shall be enveloped and apply to the entire Packing List. 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 Packing List, 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 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].

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 shall contain one X509IssuerSerial element and one X509Certificate element.

6 Asset Structure

A Packing List contains a list of assets. An asset is a set of data, such as essence or metadata. Any type of data may be referred to as an asset. Each asset shall be described by an `Asset` element as described in Figure 4. See the XML Schema declaration in Section 7 of this document for explicit type information.

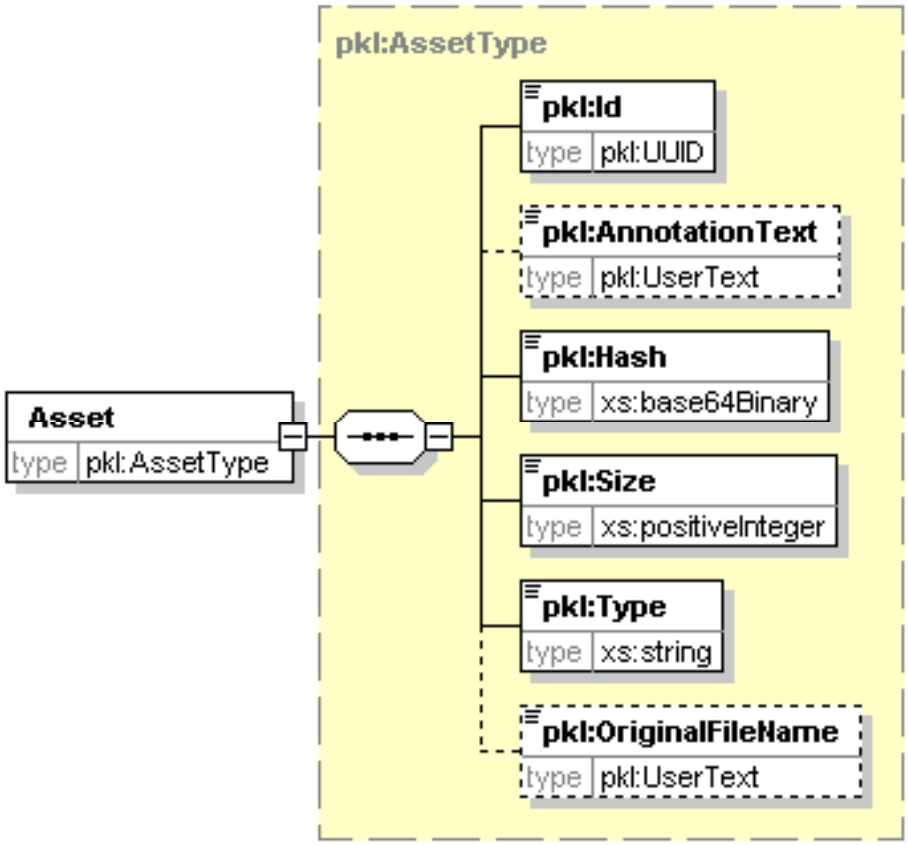


Figure 4 – Asset Structure. Dotted lines denote an optional element.

6.1 Id

The `Id` element uniquely identifies the asset for management purposes. It shall be encoded as a urn:UUID [RFC 4122], which shall be extracted from the asset where possible. The allocation of the UUID is done by the creator of the asset. The mapping of UUID values to actual asset resources is beyond the scope of this document.

6.2 AnnotationText [optional]

The `AnnotationText` element, if present, shall be a free-form, human-readable annotation associated with the asset. It is meant strictly as a displayed guidance for 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. Here are some examples:

```
<AnnotationText>The Jazz Singer – Swedish Package</AnnotationText>

<AnnotationText>The Jazz Singer – new reel 5 with outtakes -
Swedish Package</AnnotationText>
```

6.3 Hash

The `Hash` element shall be a Base64 [RFC 2045] representation of the SHA-1 message digest of the asset. SHA-1 is defined by [RFC 3174]. This message digest is used for integrity checking of the asset.

6.4 Size

The `Size` element contains the size of the asset. This size is expressed as an integer number of bytes, encoded as an `xs:positiveInteger`.

6.5 Type

The `Type` element describes the MIME type of the asset file. It is meant both as a displayed guidance for the user and as machine-interpretable information for content reception processing. MIME types are defined in [RFC2045] and [RFC2046]. The `Type` parameter shall contain any valid MIME type string as shown by example in Table 2. The MIME type strings associated with other D-Cinema Packaging structures are defined in their respective normative documents.

Table 2 – Sample Asset Types (Informative)

Asset Type Enumeration	
Type	*Description*
`text/xml`	An XML file (Compositon playlist, subtitle track file,…)
`application/mxf`	An MXF track file (picture or sound)

6.6 OriginalFileName [optional]

The `OriginalFileName` element contains the name of the file containing the asset at the time the Packing List was created. The value of this element is not constrained to any specific format. It is meant strictly as a displayed guidance for the user.

7 XML Schema

The XML Schema document presented in this section normatively defines the structure of a Packing List using a machine-readable language. While this schema is intended to faithfully represent the structure presented in the normative prose portions (Sections 4 to 6) 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.

```
<xs:schema targetNamespace="http://www.smpte-ra.org/schemas/429-8/2007/PKL"
    xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
    xmlns:xs="http://www.w3.org/2001/XMLSchema"
    xmlns:pkl="http://www.smpte-ra.org/schemas/429-8/2007/PKL"
    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"/>
  <!-- the remaining subsections are inserted here -->
</xs:schema>
```

7.1 PackingList

```
<xs:element name="PackingList" type="pkl:PackingListType"/>
  <xs:complexType name="PackingListType">
    <xs:sequence>
      <xs:element name="Id" type="pkl:UUID"/>
      <xs:element name="AnnotationText" type="pkl:UserText" minOccurs="0"/>
      <xs:element name="IconId" type="pkl:UUID" minOccurs="0"/>
      <xs:element name="IssueDate" type="xs:dateTime"/>
      <xs:element name="Issuer" type="pkl:UserText"/>
      <xs:element name="Creator" type="pkl:UserText"/>
      <xs:element name="GroupId" type="pkl:UUID" minOccurs="0"/>
      <xs:element name="AssetList">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="Asset" type="pkl:AssetType" 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>
```

7.2 Asset

```
  <xs:element name="Asset" type="pkl:AssetType"/>
  <xs:complexType name="AssetType">
    <xs:sequence>
      <xs:element name="Id" type="pkl:UUID"/>
      <xs:element name="AnnotationText" type="pkl:UserText" minOccurs="0"/>
      <xs:element name="Hash" type="xs:base64Binary"/>
      <xs:element name="Size" type="xs:positiveInteger"/>
      <xs:element name="Type" type="xs:string"/>
      <xs:element name="OriginalFileName" type="pkl:UserText" minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>
```

7.3 Misc

```
<!-- UUID -->
<xs:simpleType name="UUID">
  <xs:restriction base="xs:anyURI">
    <xs:pattern value="urn:uuid:[0-9a-fA-F]{8}-[0-9a-fA-F]{4}-[0-9a-fA-F]{4}-[0-9a-fA-F]{4}-
                       [0-9a-fA-F]{12}"/>
  </xs:restriction>
</xs:simpleType>

<!--UserText-->
<xs:complexType name="UserText">
  <xs:simpleContent>
    <xs:extension base="xs:string">
      <xs:attribute name="language" type="xs:language" use="optional" default="en"/>
    </xs:extension>
  </xs:simpleContent>
</xs:complexType>
```

Annex A (Informative)
Sample

The following Packing List sample XML structure is a valid instance of the Packing List schema. It is not-functional and meant for informative purposes only. The optional *Signature* and *Signer* elements have been omitted for the sake of clarity and to save space.

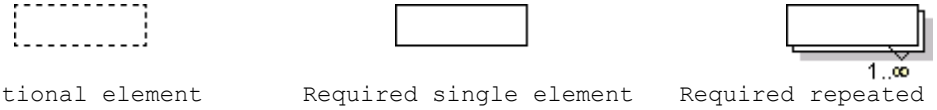
```
<?xml version="1.0" encoding="UTF-8"?>
<pkl:PackingList xmlns:pkl="http://www.smpte-ra.org/schemas/429-8/2007/PKL"
    xmlns:ds=http://www.w3.org/2000/09/xmldsig#>

  <pkl:Id>urn:uuid:c287765c-0db2-427c-9903-2d65e459c648</pkl:Id>
  <pkl:IssueDate>2005-11-02T16:39:22-00:00</pkl:IssueDate>
  <pkl:Issuer>D-Cinema Distribution Co.</pkl:Issuer>
  <pkl:Creator>Package Tool 1.0</pkl:Creator>
  <pkl:AssetList>
    <pkl:Asset>
      <pkl:Id>urn:uuid:ad03e262-820b-4bf6-9646-d64eed451c92</pkl:Id>
      <pkl:AnnotationText>Jazz Singer Reel #1 Picture</pkl:AnnotationText>
      <pkl:Hash>VtRLivH5riIVeJaaSlvJF1KW5VI=</pkl:Hash>
      <pkl:Size>15739456732</pkl:Size>
      <pkl:Type>application/x-smpte.mxf</pkl:Type>
    </pkl:Asset>
    <pkl:Asset>
      <pkl:Id>urn:uuid:bdbad39f-4967-45aa-8dd5-4b12f866e598</pkl:Id>
      <pkl:AnnotationText>Jazz Singer Reel #1 Sound</pkl:AnnotationText>
      <pkl:Hash>yH8lL/gA3y6Fnykmy+9Qmsp59g4=</pkl:Hash>
      <pkl:Size>4232864301</pkl:Size>
      <pkl:Type>application/x-smpte.mxf</pkl:Type>
    </pkl:Asset>
    <pkl:Asset>
      <pkl:Id>urn:uuid:82ba97a0-b6fc-49af-9cfe-e090f4e4f524</pkl:Id>
      <pkl:AnnotationText>Jazz Singer Composition List</pkl:AnnotationText>
      <pkl:Hash>oloZQKdmLbtuVhTr8GuQYlQdcD0=</pkl:Hash>
      <pkl:Size>5258</pkl:Size>
      <pkl:Type>text/xml</pkl:Type>
    </pkl:Asset>
  </pkl:AssetList>
</pkl:PackingList>
```

Annex B (Informative)
XML Diagram Legend

The following provides a legend for notation used in diagrams depicting XML structures.

B.1 Element symbols

In the schema design diagrams presented above in this document, only the elements are drawn. Attributes are not visible. The cardinality of the element (0..1, 1 exactly, 0..n, 1..n) is indicated by the border of the elements. Optional elements are drawn with a dashed line, required elements with a solid line. A maximum occurrence greater one is indicated by a double border.



Optional element
Min. occurrence = 0,
Max. occurrence = 1

Required single element
Min. occurrence = 1,
Max. occurrence = 1

Required repeated element
Min. occurrence = 1,
Max. occurrence = unbounded

The content model of elements is symbolized on the left and right side of the element boxes. The left side indicates whether the element contains a simple type (text, numbers, dates, etc.) or a complex type (further elements). The right side of the element symbol indicates whether it contains child elements or not:

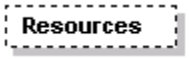
simple content

complex content

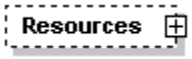
complex content
with child elements

no element content
(simple type, attributes
only, or empty element)

B.1.1 Examples



Optional single element without child elements. Minimum Occurrence = 0, Maximum Occurrence = 1, content = complex.



As above, but with child elements. The "plus" at the right side indicates the presence of one or more undisplayed child elements.

Mandatory single element. Minimum Occurrence = 1, Maximum Occurrence = 1, content = complex, no child elements (i.e. this denotes an *empty element*). The gray or green text below the element displays the xml-schema annotation associated with the element.

Mandatory multiple element containing child elements (content = complex). This element must occur at least once (Minimum Occurrence = 1) and may occur as often as desired (Maximum Occurrence = unbounded).

Mandatory single element containing simple content (e.g. text) or mixed complex content (e.g. text with xhtml markup). Minimum Occurrence = 1, Maximum Occurrence = 1, type = xsd:string (for example), content = simple. The three lines in the upper left corner are used for both text and numeric content.

B.2 Model symbols ("compositors")

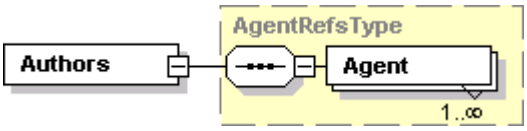
A sequence of elements. The elements must appear exactly in the sequence in which they appear in the schema diagram.

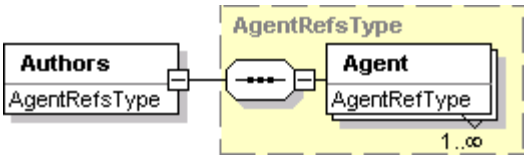
A choice of elements. Only a single element from those in the choice may appear at this position.

The "all" model, in which the sequence of elements is not fixed.

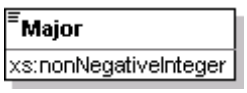
B.3 Types

If an element refers to a complex global type, the type is shown with a border.





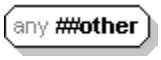
The type names of simple types are shown as well:



B.4 Model groups and references

An *element group* is a named container with one or several elements. The group of elements can be reused at multiple places in the schema. Model groups are invisible in the instance document. Model groups have been used sparingly since they do not map to a feature in object-oriented programming languages (unless they support multiple inheritance).

Import note on reading the diagrams for model groups: If the model group symbol is drawn with simple lines (i.e. not dashed), this does not imply that the elements in the model group are required. The optionality of the group depends on the optionality of elements contained in the model group. (Model groups can be made optional, e.g. to make a model group with required elements optional in some cases, but this has not been used.)



The *"any"* group is a special kind of model group. It is a placeholder for elements not defined in the schema. The "any" element defines points where the schema can be extended. After the "Any" keyword the namespace from which the elements may come is defined, for example, "##other" specifies that the extension elements may come from any namespace, except from the current schema namespace.

Element references are indicated through a link arrow in the lower left corner. They are similar to references to model groups within a schema, but instead of refining the model group, they directly refer to a single global element. The global element can then be reused in multiple places.

Annex C (Informative)
Bibliography

World Wide Web Consortium (W3C) - *RDDL - Resource Directory Description Language* J. Border and T. Bray 2002. http://www.rddl.org/

World Wide Web Consortium (W3C) - *Namespaces in XML,* http://www.w3.org/TR/REC-xml-names/

World Wide Web Consortium (W3C) - *QA Framework: Specification Guidelines, Formal Languages*, http://www.w3.org/TR/2004/WD-qaframe-spec-20041122/

World Wide Web Consortium (W3C) – *XML Schema Primer,* http://www.w3.org/

http://www.iana.org/assignments/media-types

SMPTE 429-3-2006, D-Cinema Packaging — Sound and Picture Track File

SMPTE 429-7-2006, D-Cinema Packaging — Composition Playlist