SMPTE STANDARD

Interoperable Master Format — Output Profile List

Table of Contents

Approved
December 12, 2014

Foreword

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

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

SMPTE ST 2067-100 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 Standard. 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 Output Profile List instance is an XML document that contains the instructions necessary to transform a single IMF Composition, as specified in SMPTE ST 2067-3, into deliverables tailored to one or more downstream distribution channels.

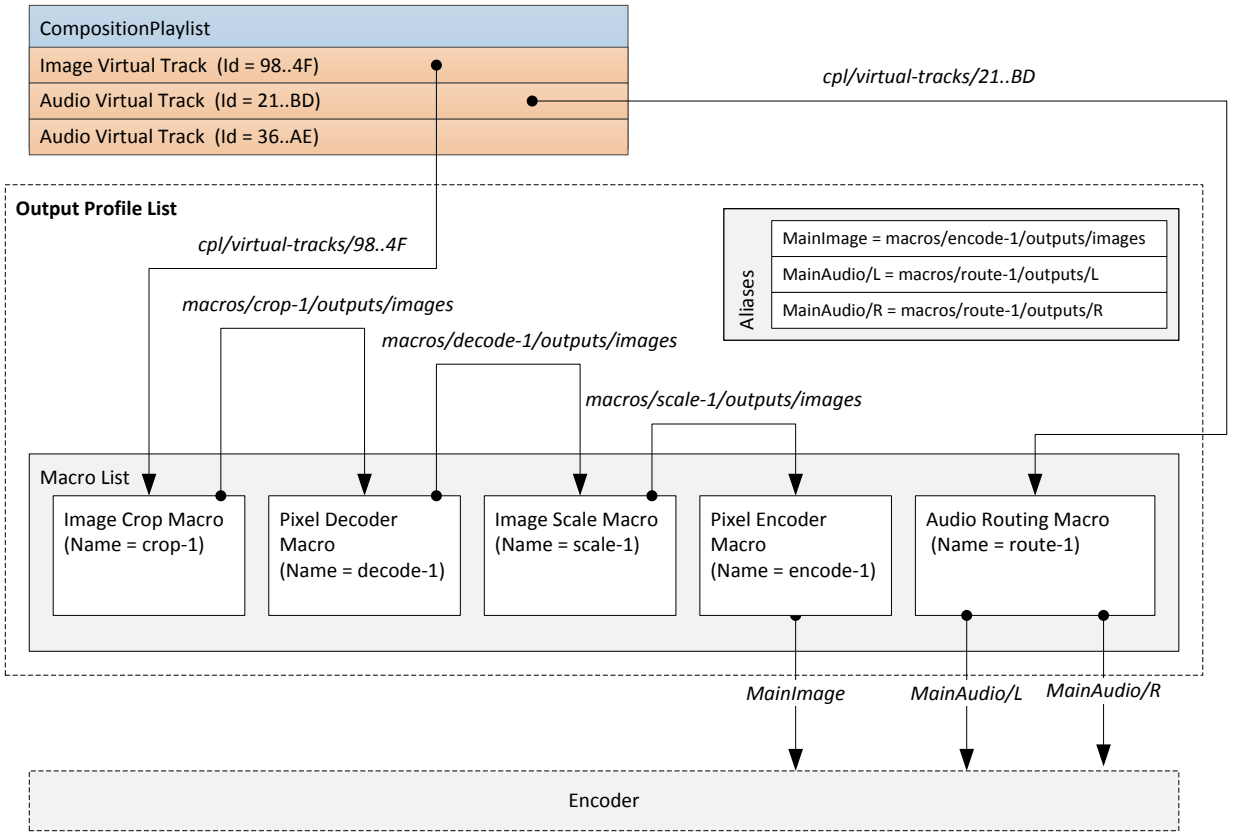

Figure 1 – Output Profile List Example. Only the first and last bytes of UUIDs are represented.

An Output Profile List consists of an ordered list of parameterized operations, called Macros, that are applied sequentially[1] to the virtual tracks of the Composition. Each Macro has a Name, which is unique within the Output Profile List instance, and exposes a number of outputs. Macros can take many forms to address the wide range of essence kinds found in a Composition, and support both low- and high-level processing, from image pixel operations to subtitle rendering.

This specification specifies the data structure (Section 5) and a processing model (Section 6) for the Output Profile List. The processing model is expressed in terms of an abstract `MacroType` structure and independently of the underlying essence kinds. This allows Macros to be defined by other specifications, whether for private or standard use, by merely extending the structure. A minimal profile of the Output Profile List, called the Simple Output Profile List, is specified in Annex B.

Essence or metadata available for processing within an Output Profile List — whether originating from the Composition, e.g. virtual tracks, or the output of a Macro — is assigned a unique name, called a Handle (Section 8). Handles are used to tie the output of one Macro to the input of another, and by external

[1] While the Output Profile List structure specifies a flat sequence of Macros, implementation can parallelize the execution of two Macros whenever the first does not depend on the output of the second.

processes, e.g. encoder. For convenience, Handles can also be aliased so that the same information, e.g. primary image output, can be referred to using a known Handle, independently of its source.

Figure 1 illustrates an example use of the Output Profile List framework. Using its handle ("`cpl/virtual-tracks/98..4F`"), an image Virtual Track from the Composition Playlist is tied into the input of an image cropping Macro named "`crop-1`". The output of this macro ("`macros/crop-1/outputs/images`") is then input into a pixel decoding Macro, which transforms the cropped image into a form appropriate for processing by a scaling Macro. The output of the scaling macro is then converted back to a common pixel encoding format, e.g. 8-bit RGB, by a pixel encoding Macro ("`encode-1`"). The output of the latter is aliased to "`MainImage`", essentially hiding internal Macro names from external processing services. Separately, an audio Virtual Track is processed by an audio routing Macro that exposes two outputs, one for each of the left and right channel of the Composition. These two outputs are similarly aliased.

While this specification generally leaves the definition of Macros to other specifications, Annex B specifies the Preset Macro. The Preset Macro allows the creation of a special kind of Output Profile List, the Simple Output Profile List, where all processing is reduced to a single opaque URI.

1 Scope

An Output Profile List (OPL) defines the transformation of selected virtual tracks of a single Interoperable Master Format (IMF) Composition into deliverables appropriate for downstream distribution channels. This transformation consists of a sequence of parameterized steps, called Macros. This document specifies the basic Output Profile List structures and processing model, independently of the underlying essence kinds.

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

Note: All references in this document to other SMPTE documents use the current numbering style (e.g. SMPTE ST 433:2008) although, during a transitional phase, the document as published (printed or PDF) may bear an older designation (such as SMPTE 433-2008). Documents with the same root number (e.g. 433) and publication year (e.g. 2008) are functionally identical.

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.

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

SMPTE ST 2067-3:2012, Interoperable Master Format — Composition Playlist

IETF (2005, January). RFC 3986 — Uniform Resource Identifier (URI): Generic Syntax

IETF (2005, December). RFC 4288 — Media Type Specifications and Registration Procedures

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.

4 General

4.1 Instance

An Output Profile List instance shall be an XML document, as specified in W3C Extensible Markup Language, which consists of a single `OutputProfileList` element.

4.2 Schema

An Output Profile List instance shall conform to the inline XML schema definitions (as specified in W3C XML Schema Part 1: Structures and W3C XML Schema Part 2: Datatypes) found in this specification. In the event of a conflict between schema definitions and the prose, the prose shall take precedence.

The XML schema root element shall be as defined in Table 1.

Table 1 – XML Schema root element definition

```
<xs:schema targetNamespace="http://www.smpte-ra.org/schemas/2067-100/2014"
                   xmlns:opl="http://www.smpte-ra.org/schemas/2067-100/2014"
                  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="OutputProfileList" type="opl:OutputProfileListType"/>
<!-- schema definitions found in this document excluding this one -->
</xs:schema>
```

Note 1: The prefix associated with each namespace in an XML Schema definition document is arbitrary, and an Output Profile List instance can use other prefix value for the same namespace.

Note 2: The XML Schema definitions found in this specification include elements specified in W3C XML Digital Signature and SMPTE ST 433.

4.3 Character Encoding

An Output Profile List instance shall use UTF-8 encoding, as specified in W3C Extensible Markup Language (XML).

4.4 Media Type

The media type, as defined in IETF RFC 4288, of an Output Profile List instance shall be `text/xml`.

4.5 Versioning

The namespace of the `OutputProfileList` element of an Output Profile List instance allows implementations to unambiguously identify the defining specification of the Output Profile List instance.

Output Profile List instances conforming to specifications that modify the schema definitions or semantics of the elements defined herein in an incompatible way, including future versions of this specification, shall use a different namespace for the `OutputProfileList` element and no two incompatible schemas shall use the same namespace for the `OutputProfileList` element.

5 Structures

5.1 General

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

5.2 OutputProfileListType

5.2.1 General

The `OutputProfileListType` type shall be specified by Table 2.

Table 2 – OutputProfileListType schema definition

```
<xs:complexType name="OutputProfileListType">
  <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="CompositionPlaylistId" type="dcml:UUIDType"/>
    <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="AliasList">
      <xs:complexType>
        <xs:sequence>
          <xs:element maxOccurs="unbounded" minOccurs="0" name="Alias">
            <xs:complexType>
              <xs:simpleContent>
                <xs:extension base="opl:HandleType">
                     <xs:attribute name="handle" type="opl:HandleType"
                           use="required"/>
                </xs:extension>
              </xs:simpleContent>
            </xs:complexType>
          </xs:element>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
    <xs:element name="MacroList">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="Macro" type="opl:MacroType" 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>
```

5.2.2 Id

The `Id` element uniquely and globally identifies the Output Profile Playlist instance.

Any two Output Profile Playlist instances may have identical `Id` values if and only if the two Output Profile Playlist instances are identical.

5.2.3 Annotation

The `Annotation` element shall be a free-form, human-readable annotation describing the Output Profile Playlist. It is intended strictly for display to the user.

5.2.4 IssueDate

The `IssueDate` element shall indicate the time and date at which the Output Profile Playlist was issued.

5.2.5 Issuer

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

5.2.6 Creator

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

5.2.7 CompositionPlaylistId

The `CompositionPlaylistId` shall be equal to the `Id` element of the Composition Playlist instance onto which the Output Profile List operates.

5.2.8 ExtensionProperties

The `ExtensionProperties` element contains an unordered list of `xs:any` elements, which may be used by applications to add descriptive metadata to the Output Profile List.

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

5.2.9 AliasList

Each `Alias` element of the `AliasList` element defines a synonym for the Handle specified in the `handle` attribute – see Section 6.

The `handle` attribute shall be a Handle as specified in Section 6.

No two `Alias` elements shall have the same value.

5.2.10 MacroList

The `MacroList` element shall contain an ordered list of `Macro` elements.

Multiple `Macro` elements may have the same type but no two `Macro` elements shall have the same `Name` value.

Macro elements shall conform to the constraints of Section 6.

5.2.11 Signer

The `Signer` element shall uniquely identify the entity that digitally signed the Output Profile List.

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

If X.509 certificates are used as specified in W3C XML Digital Signature and Processing, then the Signer element shall contain one `X509Data` element containing one `X509IssuerSerial` element, which uniquely identifies the certificate used to sign the Output Profile List.

5.2.12 Signature

The `Signature` element shall contain a digital signature authenticating the Output Profile List.

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

The digital signature shall be enveloped, as specified in W3C XML Digital Signature and Processing, and apply to the `OutputProfileList` element. The signature shall be generated by the signer identified by the `Signer` element.

5.3 MacroType

5.3.1 General

The `MacroType` type shall be specified by Table 3.

Table 3 – MacroType schema definition

```
<xs:complexType name="MacroType" abstract="true">
   <xs:sequence>
    <xs:element name="Name" type="opl:MacroNameType"/>
    <xs:element name="Annotation" type="dcml:UserTextType" minOccurs="0"/>
   </xs:sequence>
</xs:complexType>
```

5.3.2 Name

The `Name` element is the Name of the Macro instance and uniquely identifies it.

5.3.3 Annotation

The `Annotation` element shall be a free-form, human-readable annotation describing the Macro instance.

It is intended strictly for display to the user.

5.4 HandleType

The `HandleType` type represents Handles as specified in Section 8.

The `HandleType` type shall be specified by Table 4.

Table 4 – HandleType schema definition

```
<xs:simpleType name="HandleType">
    <xs:restriction base="xs:anyURI"/>
</xs:simpleType>
```

5.5 MacroNameType

The `MacroNameType` type represents the Name of a Macro.

The `MacroNameType` type shall be specified by Table 5.

Table 5 – MacroNameType schema definition

```
<xs:simpleType name="MacroNameType">
  <xs:restriction base="xs:string">
    <xs:pattern value="[a-zA-Z][a-zA-Z0-9-]*"/>
  </xs:restriction>
</xs:simpleType>
```

6 Macro Constraints

Each `Macro` element shall be derived, directly or indirectly, by extension from the `MacroType` abstract type (see Section 5.3).

The type of a `Macro` element shall unambiguously identify the specification that specifies its operation.

The defining specification of a `Macro` element shall:

- specify all inputs required by a Macro instance, allowing Macro ordering as specified in Section 7;
- specify a Handle (in the context of the Macro instance per Section 8.2.5) for Macro output that is exposed; and
- specify the process by which Macro inputs are transformed to outputs.

7 Processing Model

Each `Macro` present in the `MacroList` element shall be executed in full and in the order it appears in the element.

A first `Macro` instance that references the output of a second `Macro` instance shall not appear in the `MacroList` before the second `Macro` instance.

8 Handles

8.1 General

Essence or metadata available for processing within an OPL can be unambiguously referenced by a Handle. In particular, Handles are used to tie the output of one Macro to the input of another, and by external processes, e.g. encoder, to reference essence for further processing.

A Handle shall be a relative-path reference URI as specified in IETF RFC 3986.

Handles are case-sensitive.

A Handle shall always be specified within a context, and, within a context, a given Handle shall not reference two distinct entities.

Annex A specifies a grammar to simplify the specification of Handle syntax.

The following definitions are provided for convenience:

```
<hexdig> ::= [a-fA-F0-9]
<uuid> ::= <hexdig>{8} ("-" <hexdig>{4}){3} "-" <hexdig>{12}
<alpha> ::= [a-zA-Z]
<digit> ::= [0-9]
<symbol> ::= <alpha> (<alpha> | <digit> | "-")*
```

8.2 Handles within the Context of an Output Profile List Instance

8.2.1 General

The following is a non-exhaustive list of Handle definitions. Other specifications may define additional Handles, both inside and outside the context of an Output Profile List.

8.2.2 Composition Playlist Instance

The Composition Playlist instance shall be associated with a Handle conforming to the `<cpl-handle>` syntax:

```
<cpl-handle> ::= "cpl"
```

8.2.3 Composition Playlist Virtual Tracks Instances

A virtual track within the Composition Playlist instance shall be associated with a Handle conforming to the `<virtual-track-handle>` syntax:

```
<virtual-track-handle> ::= <cpl-handle> "/virtual-tracks/" <virtual-track-id>
<virtual-track-id> ::= <uuid>
```

where `<virtual-track-id>` is the UUID value of the `Id` element of the Virtual Track (see SMPTE ST 2067-3).

EXAMPLE: `cpl/virtual-tracks/ddcf49e4-dc5a-49ef-9eb5-55a2fd55c3cc`

8.2.4 Macro Instance

A Macro instance shall be associated with a Handle conforming to the `<macro-handle>` syntax:

```
<macro-handle> ::= "macros/" <macro-name>
<macro-name> ::= <symbol>
```

where `<macro-name>` is the value of the `Name` element of the Macro instance.

EXAMPLE: `macros/image-scaler-1`

8.2.5 Macro Instance Output

The output of a Macro instance shall be associated with a Handle conforming to the `<macro-output-handle>` syntax:

```
<macro-output-handle> ::= <macro-handle> "/outputs/" <local-output-handle>
```

where

- `<macro-handle>` is the Handle of the Macro instance; and
- `<local-output-handle>` is a Handle within the context of the Macro instance and uniquely identifies the output.

The Handle associated with the output of a Macro instance shall be specified in the specification defining the Macro instance.

EXAMPLE: Given a Macro named `"image-scaler-1"`, which exposes an output with a Handle of `"images"` in the context of the Macro instance, the Handle of the output within the context of the Output Profile List instance would be `"macros/image-scaler-1/outputs/images"`.

8.3 Aliases

The Handle associated with an `Alias` element (see Section 5.2.9) shall conform to the `<alias-handle>` syntax:

```
<alias-handle> ::= "alias/" <target-handle>
```

where `<target-handle>` is the value of the `Alias` element.

The Handle `<alias-handle>` shall be a synonym of the Handle specified by the `handle` attribute of the `Alias` element.

EXAMPLE: given the `Alias` element `"<Alias handle='macros/image-scaler-1/outputs/images'>MainImage</Alias>"`, the Handle `"alias/MainImage"` is synonymous with the Handle `"macros/image-scaler-1/outputs/images"`.

Annex A Handle Grammar (Normative)

Handles are specified using the formal grammar of XML as specified in Section 6 of Extensible Markup Language (XML) 1.0, with the following additions.

`expression{n}` matches exactly `n` repetitions of `expression`.

`expression{n,m}` matches between `n` and `m` (inclusive) repetitions of `expression`.

Annex B Preset Macro and Simple OPL (Normative)

A Preset Macro instance shall be a Macro instance with the type `PresetMacroType` defined in Table B.1.

Table B.1 – PresetMacroType schema definition

```
 <xs:complexType name="PresetMacroType">
   <xs:complexContent>
    <xs:extension base="opl:MacroType">
     <xs:sequence>
      <xs:element name="Preset" type="xs:anyURI"/>
     </xs:sequence>
    </xs:extension>
   </xs:complexContent>
</xs:complexType>
```

An Output Profile List instance that contains a Preset Macro instance shall not contain any other Macro instances. Such an Output Profile List instance is called a Simple OPL.

The value of the `Preset` element of the Preset Macro instance shall fully specify the processing applied to the Output Profile List.

This specification does not define the operation of the Preset Macro, which is left to private arrangements between sender and recipients of the Output Profile List instance.

A Preset Macro instance does not expose outputs.

Note: If a single private Macro instance needs to be inserted in an otherwise standard sequence of Macro instances, the private Macro instance can simply be derived from `MacroType` and no Preset Macro is required.

Annex C Consolidated Schema (Informative)

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

st2067-100a-2014.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 D Sample Instance (Informative)

No sample Output Profile List instance is included.