Using Common Media Manifest with Interoperable Media Format (IMF)

CONTENTS

1 Introduction 4
1.1 Background 4
1.2 Document Organization 4
1.3 Document Naming and Conventions 4
1.4 Normative References 5
2 Referencing IMF CPL Virtual Tracks 6
2.1 General 6
1.1 TrackReference 6
1.2 TrackIdentifier 6
1.3 ContainerReference 7
2 Referencing OPL Macro Outputs 7
2.1 General 7
2.2 TrackReference 7
2.3 TrackIdentifier 7
2.4 ContainerReference 7
3 Referencing an IMF Container 7
3.1 ContainerLocation 7
3.2 ContainerIdentifier 8
3.3 ParentContainer 8
4 IMF Mapped File Set 8
4.1 General 8
4.2 Asset Map Document Location 8
4.3 Asset Map Document Encoding 8
4.4 Asset Location and Path Constraints 8
4.5 Retrieving Resources 9

NOTE: No effort is being made by the Motion Picture Laboratories to in any way obligate any market participant to adhere to this specification. Whether to adopt this specification in whole or in part is left entirely to the individual discretion of individual market participants, using their own independent business judgment. Moreover, Motion Picture Laboratories disclaims any warranty or representation as to the suitability of this specification for any purpose, and any liability for any damages or other harm you may incur as a result of subscribing to this specification.

REVISION HISTORY

Version	Date	Description

1 INTRODUCTION

1.1 Background

Common Manifest allows user experiences to be assembled from individual assets, e.g. a video track, audio track, etc.

The Interoperable Master Format (IMF) Framework facilitates the management and processing of multiple Compositions (airline edits, special edition …) of the same high-quality finished work (feature, episode, trailer, etc.). The playback timeline of each Composition is controlled by a Composition Playlist (CPL), which is specified in SMPTE ST 2067-3 and defines a Virtual Track for each essence kind, e.g. video, audio, etc.

An IMF Output Profile List (OPL), which is specified in SMPTE ST 2067-100, can then be used to transform (scaled, cropped, etc.) Virtual Tracks to meet the specific needs of downstream distribution channels.

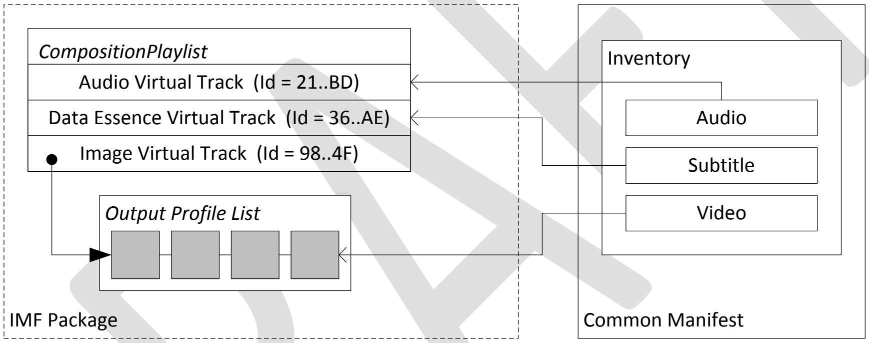
CompositionPlaylist
Audio Virtual Track (Id = 21..BD)
Data Essence Virtual Track (Id = 36..AE)
Image Virtual Track (Id = 98..4F)
Output Profile List
IMF Package
Inventory
Audio
Subtitle
Video
Common Manifest

Figure 1. Referencing an IMF Package from Common Manifest.

As illustrated in Figure 1, this document allows Common Manifest to reference IMF CPL Virtual Tracks and IMF OPL Macro Outputs. The IMF files can be either local or accessed through HTTP or HTTPS.

1.2 Document Organization

TBS

1.3 Document Naming and Conventions

This specification uses 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`.

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]
```

1.4 Normative References

SMPTE ST 2067-2:2013, Interoperable Master Format – Core Constraints

SMPTE ST 2067-3:2013, Interoperable Master Format – Composition Playlist

IETF RFC 3986, Uniform Resource Identifier (URI): Generic Syntax

SMPTE ST 429-9:2014, D-Cinema Packaging —Asset Mapping and File Segmentation

SMPTE ST 2067-100:2014, Interoperable Master Format – Output Profile List

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

MovieLabs (March 11, 2014), Common Media Manifest Metadata v0.7f

IETF RFC 2616, Hypertext Transfer Protocol — HTTP/1.1

IETF RFC 5246, The Transport Layer Security (TLS) Protocol, Version 1.2.

2 REFERENCING IMF CPL VIRTUAL TRACKS

2.1 General

A Common Manifest Inventory Asset that references an IMF CPL Virtual Track shall conform to this Section.

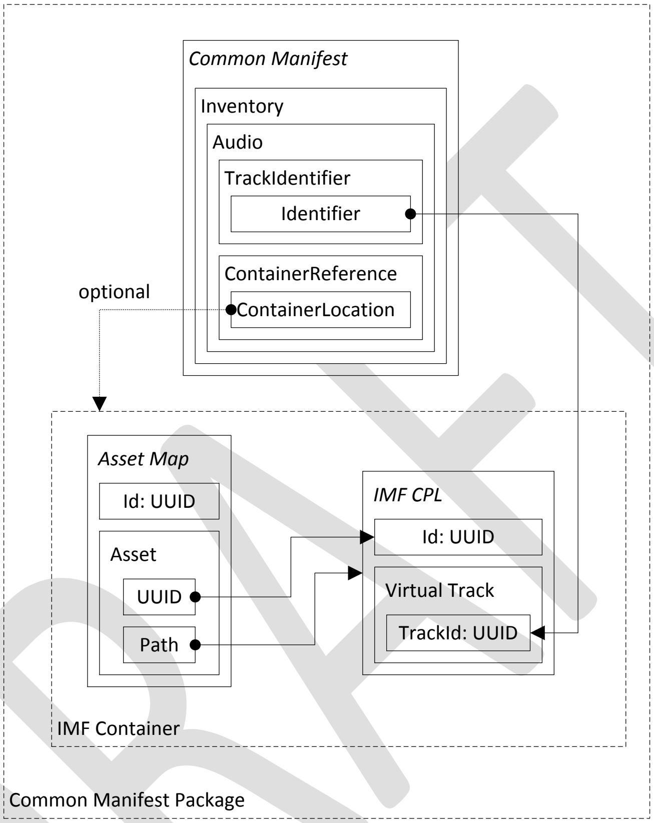

Figure 2. Referencing an IMF CPL Virtual Track.

1.1 TrackReference

The `TrackReference` element is not used and should not be present.

1.2 TrackIdentifier

The `TrackIdentifier/Namespace` element shall be `"smpte:imf"`.

The `TrackIdentifier/Identifier` element shall conform to the `<track-id>` syntax:

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

where `<cpl-id>` shall be the `Id` of the Composition Playlist (see Section 6.1.1 of SMPTE ST 2067-3) and `<virtual-track-id>` shall be equal to the `TrackId` of the CPL Virtual Track (see Section 6.9.3 of SMPTE ST 2067-3).

The `TrackIdentifier/Location` element is not used and should not be present.

1.3 ContainerReference

If present, the `ContainerReference` element shall conform to Section 3.

2 REFERENCING OPL MACRO OUTPUTS

2.1 General

A Common Manifest Inventory Asset that references an OPL Macro Output shall conform to this Section.

2.2 TrackReference

The `TrackReference` element is not used and should not be present.

2.3 TrackIdentifier

The `TrackIdentifier/Namespace` element shall be `"smpte:imf"`.

The `TrackIdentifier/Identifier` element shall conform to the `<track-ref>` syntax:

```
<track-ref> ::= "opls/" <opl-id> "/" <macro-output-handle>
<opl-id> ::= <uuid>
```

where `<opl-id>` is the `Id` of the Output Profile List and `<macro-output-handle>` is the Macro Instance Output Handle, as specified in SMPTE ST 2067-100.

The `TrackIdentifier/Location` element is not used and should not be present.

2.4 ContainerReference

If present, the `ContainerReference` element shall conform to Section 3.

3 REFERENCING AN IMF CONTAINER

3.1 ContainerLocation

The `ContainerLocation` element shall reference the Asset Map document of a Mapped File Set that conforms to Section 4, using either:

- A relative-path reference, without query or fragment component, as specified in RFC 3986. The relative-path reference shall be resolved relative to the URI of the Common Manifest document.
- An absolute-URI as specified in RFC 3986. One of the following schemes shall be used: "http" or "https".

The Mapped File Set shall include all assets necessary to process the Common Manifest Inventory Asset.

3.2 ContainerIdentifier

The `ContainerIdentifier` element is not used and should not be present.

3.3 ParentContainer

The `ParentContainer` element is not used and should not be present.

4 IMF MAPPED FILE SET

4.1 General

IMF assets are delivered using the Asset Map mechanism specified in SMPTE ST 429-9. The following defines a Mapped File Set (see Section 9 of SMPTE ST 429-9) for use with Common Manifest deliveries.

4.2 Asset Map Document Location

The Asset Map document shall be identified by an absolute-URI whose last path segment is equal to "`ASSETMAP.xml`".

4.3 Asset Map Document Encoding

The Asset Map document shall be encoded using UTF-8 as specified in W3C Extensible Markup Language 1.0

4.4 Asset Location and Path Constraints

Each Path element of the Asset Map shall be a relative-path reference as specified in RFC 3986. No query or fragment component shall be present.

Each Asset referenced by the Asset Map shall be identified by an absolute-URI constructed by resolving, as specified in RFC 3986, its Path element relative to the Asset Map URI.

Every location traversed by this resolved relative-path reference shall be at the same or deeper hierarchical level than the Asset Map URI.

Each path segment, as specified in IETF RFC 3986, shall consist of characters from the set `a-z`, `A-Z`, `0-9`, `"-"` (dash), `"_"` (underscore) and `"."` (period). No segment shall have more than 128 characters, and the value of the Path element shall not exceed 1024 characters in length.

A Path element value shall have no more than 10 segments.

No two paths in an Asset Map shall have identical values, regardless of case.

4.5 Retrieving Resources

The resources identified by a URI using the `"http"` or `"https"` schemes shall be retrieved as the response to a `GET` request to the URI.