Media Manifest Delivery Core

CONTENTS

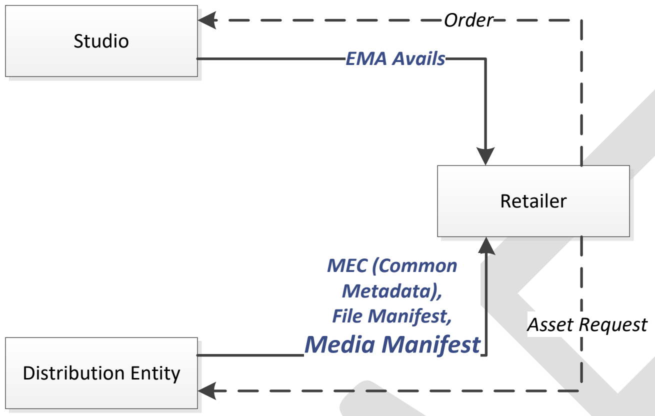
REVISION HISTORY

Version	Date	Description
1.0	January 4, 2016	First Release
2.0		Add TV

1 INTRODUCTION

The Entertainment Merchant's Association (EMA), the Digital Entertainment Group (DEG) and MovieLabs have defined a subset of MovieLabs' Media Manifest for use in Delivery. This core subset is designed for simple application of the Manifest.

The Media Manifest is part of an overall delivery workflow that is supported by complementary specifications that can optionally be used in conjunction with this specification.



Also relevant to this specification is the Media Entertainment Core (MEC) definition that defines metadata structure.

The specification and best practices for the Media Manifest can be found at www.movielabs.com/md/manifest. The full specification supports more complex delivery use cases, interactivity and other applications.

1.1 Document Organization

This document is organized as follows:

1. Introduction—Background, scope and conventions
2. Media Manifest Core –Definition of MMC
3. Manifest Use Cases – MMC Manifest use cases with examples
4. Metadata Use Casts – Metadata use cases with examples

1.2 Document Notation and Conventions

The document uses the conventions of Common Metadata [CM].

1.3 Normative References

[Manifest]	TR-META-MMM MovieLabs Media Manifest Metadata, v1.~~5~~8, http://www.movielabs.com/md/manifest

[CM]	TR-META-CM MovieLabs Common Metadata, v2.~~3~~7, http://www.movielabs.com/md/md
[EIDR-TO]	*EIDR Technical Overview*, November 2010. http://eidr.org/technology/#docs
[MEC]	*Media Entertainment Core (MEC) Metadata, version 2.~~4~~7. www.movielabs.com/md/mec and http://entmerch.org/programsinitiatives/ema-metadata-structure.*
[AVAILS]	*Entertainment Merchant's Association (EMA) Avails. www.movielabs.com/md/avails*

All Common Metadata and Media Manifest references are included by reference.

1.4 Informative References

[MMDelivery]	BP-META-MMMD, Using Media Manifest, File Manifest and Avails for file Delivery (Best Practice), v1.1, www.movielabs.com/md/manifest
[GoogleHelp]	Google Play help site. https://support.google.com/moviestvpartners/answer/6154374

1.5 XML Namespaces

This document defines specific use of the 'manifest' namespace.

'manifest' builds on the 'md' Common Metadata corresponding with Common Metadata [CM].

1.6 Identifiers

Identifiers must be universally unique. Recommended identifier schemes may be found in Common Metadata [CM] and in DECE Content Metadata [DECEMD].

The use of Entertainment Identifier Registry identifiers (www.eidr.org) is strongly encouraged. Please see [EIDR-TO].

Best practices for identifiers can be found ~~in Best Practices for Delivery [MMDelivery].~~on www.movielabs.com/md/practices.

1.7 Status

This specification is completed and ready for implementation. Although tested, we anticipate that additional implementation experience will yield recommendation for changes. Implementers should anticipate one or more revisions. Reasonable measures will be taken to ensure changes are backwards compatible. See Backwards Compatibility Best Practices in [CM]

2 MEDIA MANIFEST DELIVERY CORE

The section defines the Media Manifest Core (MMC) subset of the Media Manifest. It constrains the Media Manifest to a manageable 'core' subset for implementation.

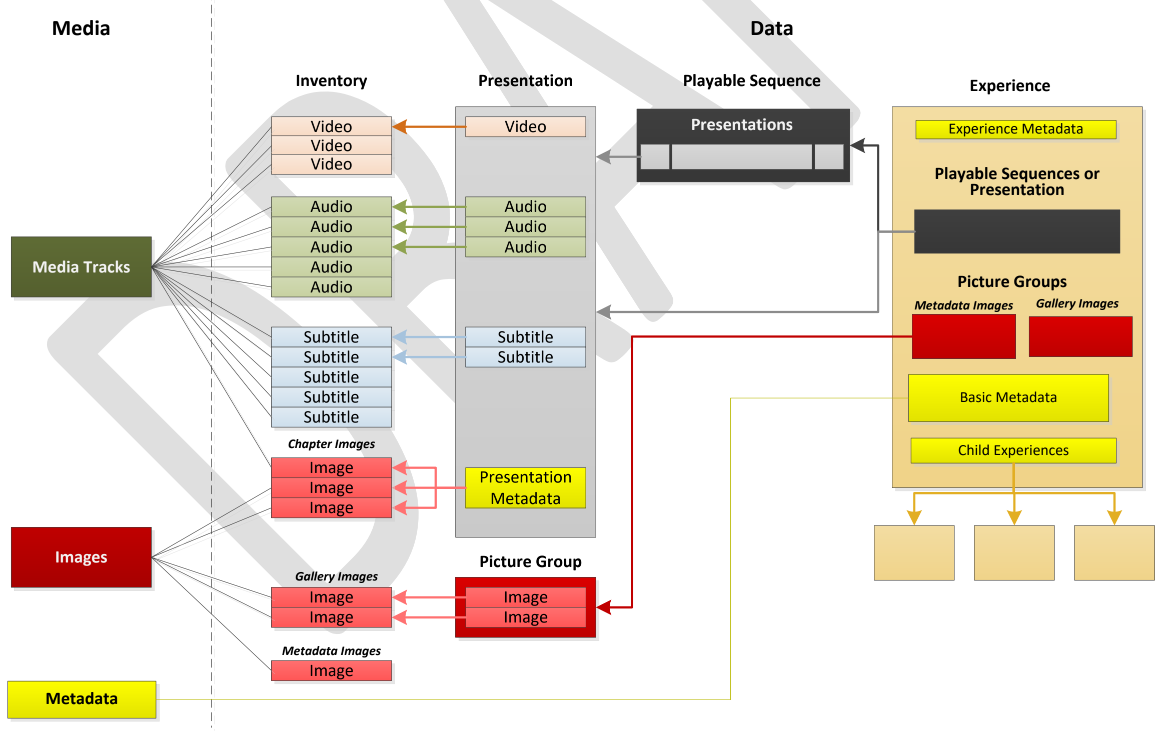
The rules for what must be included and how it is encoded is in this section and its references. Metadata is based on the Media Entertainment Core (MEC) as defined in [MEC].

The use of the Core does not preclude the use of additional Manifest features, although implementations need not support them. Additional data in the Manifest not needed in a given implementation can simply be ignored.

The document is written towards the version of MMC and MEC in References. It should be forward compatible to future versions.

2.1 Media Manifest derived types

Media Manifest [Manifest] includes elements ~~provides~~providing a layered model of information that abstracts various portions of content delivery. The following illustrates the subset of the Media Manifest used in the Media Manifest Core.



This specification uses the following abstractions:

- **Media** – Video, audio, subtitles, images, applications, etc. This specification defines neither ~~the~~ media files (i.e., video, audio, subtitles, images, etc.) nor ~~its delivery.~~the mechanism of delivering those file. These processes should not be impacted by the use of the Manifest.

- **Inventory** – The inventory describes (metadata) and points to media. ~~I~~It references down to individual tracks, images, etc. This provides the necessary abstraction to the media which, as mentioned above, can be in any format.
- **Grouping** – The manifest groups content that is related at playback. Presentation and Picture support are required; while Applications and Text are optional.
  - Presentation – Tracks that play together
  - Picture – Images that are related. This includes both metadata images (i.e., cover/key art). In other applications, this includes image galleries.
- **Experience** – Connects related content (e.g., trailers to a movie)

All mandatory elements and attributes must be included. Any optional elements may be included. The following elements and attributes are required for MMDC usage, regardless of whether they are optional, except as noted.

The following table uses the following conventions:

- Structure is given by table indentation. Parent level elements to the left.
- Attributes begin with '@'. For example, @ContentID refers to the ContentID attribute

2.1.1 Simplified Delivery Models

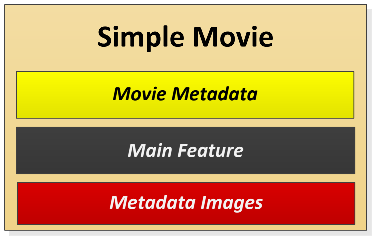
MMC is designed to make simple delivery simple. Consequently, the Core uses a constrained Experience model. This simplification comes at the expense of more complex content delivery.

While the general Manifest can support complex structure, the MMC supports the simplest and most ~~common~~frequently used content models: Movie (single title) and Episodic. They are described in the following sections.

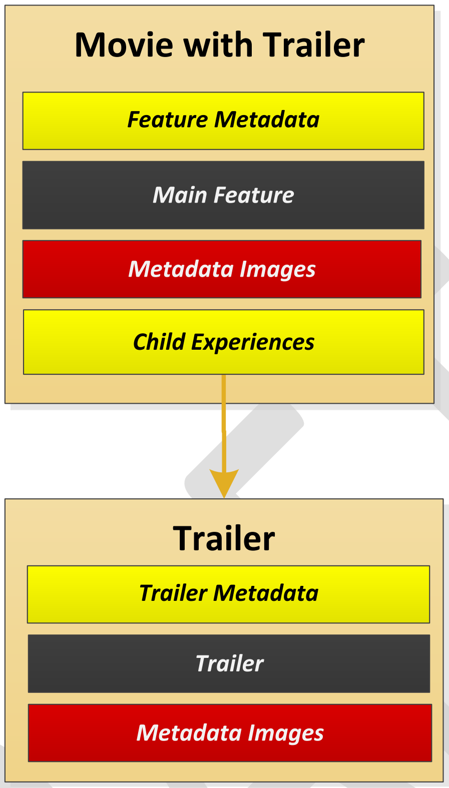
2.1.1.1 Movie Delivery Model

The movie delivery model allows the delivery of a movie, trailers and limited extras. All content is in a single Experience.

The following example show a simple movie Experience with references to metadata, metadata images and a main feature. Note that Metadata images will be in any Experience, but are not always shown in these illustrations.

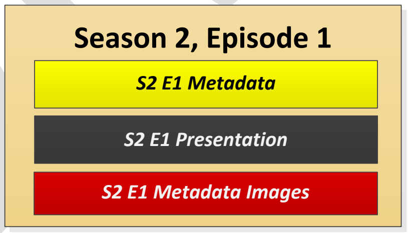


Following is more complex Experience with references to metadata, metadata images, the main feature, a trailer, a featurette (making-of), a production gallery and a couple of apps.

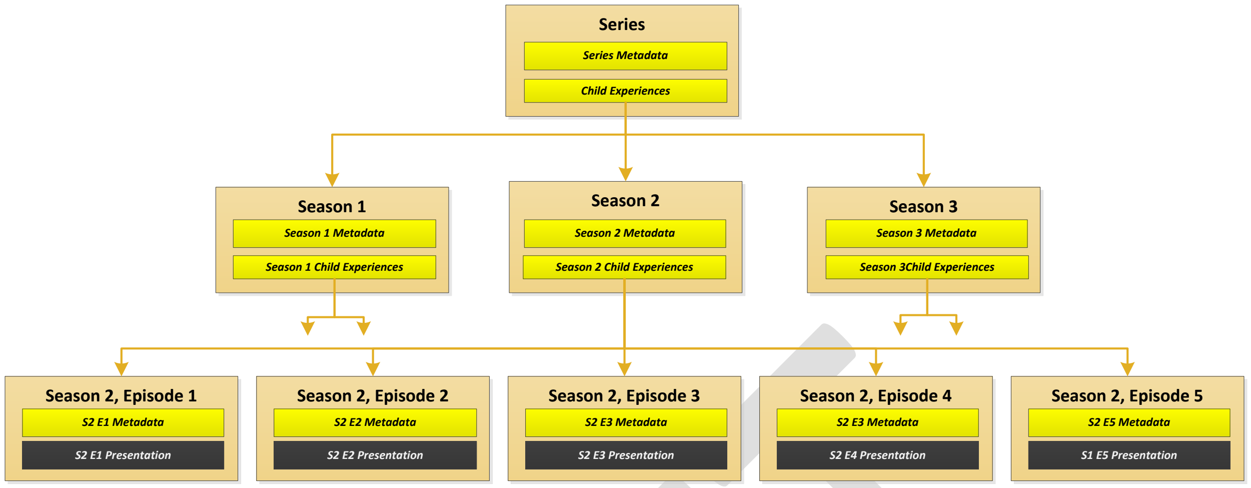


2.1.1.2 Episodic Delivery Model

Episodic content (e.g., television) handles episodes ~~like movies.~~in a manner similar to the handling of movies described in the previous section. The following shows a simple episode.



Episodes are grouped into seasons then series (or miniseries or collections) by making them children of the grouping object. The following shows a multi-season TV series. Note that metadata image references are included in each Experience, but not shown in the illustration:



2.1.2 Manifest Usage

Note that required child elements are not listed individually as they are assumed to be included.

This practice does not preclude the use of additional elements and attributes. However, this occurs by agreement between parties and is not covered by the Core.

Element or Attribute	Usage Rules
MediaManifest-type	Required
@ManifestID	Required
updateNum	Shall be included if the record is an update (i.e., not the first record distributed)
Compatibility	Required. Compatibility/Profile="MMC-1"
Inventory	Required For subordinate objects, sufficient information is required to identify the referenced media. This depends on ~~how media is delivered. For example, if a file is delivered then ContainerReference is required. If track within a container is ambiguous, then TrackReference or TrackIdentifier must be provided.~~the media files and how they are delivered.
Audio	Required for each track of this type
Video	Required for each track of this type
Subtitle	Required for each track of this type

		Image		Required for each image
		Interactive		Required for each application
		Ancillary		Required for each ancillary track (e.g., Dolby Vision)
		Metadata		Required
		TextObject		Required if text objects are included
	Presentations/Presentation			One Presentation instance is required for each set of conformed tracks.
		TrackMetadata		Required.
			TrackSelectionNumber	Required
			Video|Audio|Subtitle|Ancillary TrackReference	One instance required for each track in Presentation
	PlayableSequence			Optional, specifically intended in the Core for Dub cards.
	PictureGoups/PictureGroup			At least one PictureGroup is required for metadata images. Additional PictureGroups are optional.
	AppGroups/AppGroup			Optional
	TextGroups/TextGroup			Optional
	Experiences/Experience			First Experience instance is top-level Experience. All other Experience instance must be referenced as descendants of this Experience. See Delivery Model in Section 2.1.1.
		@ExperienceID		Required
		@updateNum		Required of Experience has been updated.
		Language, ExcludedLanguage, Region and ExcludedRegion		Optional. Typically handled by Avails.
		ContentID		Required. Most reference metadata in Inventory. Note that BasicMetadata is not allowed.
		Audiovisual		Included if this Experience references this object. At least one must be present in an Experience object.
		App		Included if this Experience references this object
		Gallery		Included if this Experience references this object

		PictureGroupID	Must be included if the ContentID references metadata with images. Additional instances are to be Included if this Experience references this object.
		TextGroupID	Included if this Experience references this object
		TimedSequenceID	Included if this Experience references this object
		ExperienceChild	Included if Experience has child Experiences.
	TimedEventSequences		Optional
	ALIDExerienceMaps/ ALIDExperienceMap		Required. Only one Experience (top-level Experience) can be referenced.

2.2 Optional Extensions to Core

Optional elements and attribute usage not specifically stated above are still allowed in a Core manifest. However, they should only be if both the sending and receiving parties agree to use these features.

This section describes some relatively simple extensions. Additional cases are provided in the best practices for delivery [Delivery]

2.2.1 Optional Pre- and Post-roll material, Dub Cards

Media Manifest offers a mechanism to allow antipiracy, rating and dub cards to be separately authored and played in sequence with the main title.

It some cases it is desirable to have material that plays before or after the primary video. Typically, antipiracy notices and ratings play before, and dub cards play after. There are two ways to implement this pre- and post-roll content. This first is to construct a video that includes everything. There are two disadvantages to this approach: a distinct asset must be created for each region and possibly language, and a player cannot dynamically select dub cards based on language being played.

When playing content in sequence, the PlayableSequence mechanism is used. A PlayableSequence is constructed with sequenced Presentation references (negative sequence numbers for content to play before the main title and positive sequence numbers for content that follows). When items, particularly dub cards, have the same sequence number, the player choose one of them based on the selected audio track being played.

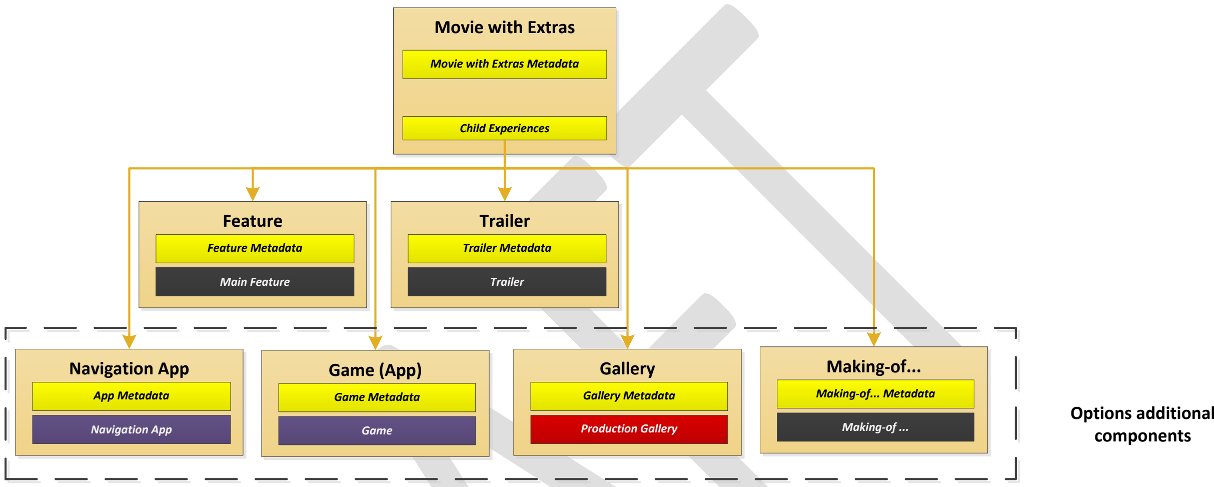
For example, an antipiracy card would have a sequence -2, a ratings card would be -1, the main feature would be 0, and all the dub cards would be 1. The player would play in sequence order the antipiracy card, the ratings card, the main feature and the card with Clip/@audioLanguage matching the selected audio language.

2.2.2 Optional Bonus Material

This section is included as an optional extension to MMC, illustrating how bonus material would be handled in a MMC Manifest.

The Manifest structure supports the inclusion of additional media assets referenced by the Manifest. Following is an example of Manifest usage with bonus material.

When adding bonus material to the Core Manifest, continue to use the constraints of the Core specified above.



2.3 Identifiers

The identifier mechanism is very flexible, but understanding a few principles and following a few rules keeps things simple. First a reminder, IDs are formatted as follows: "`md:`"<type> "`:`"<scheme>"`:`"<SSID>. For example, `md:experienceid:eidr-s:5EE7-A973-819A-DC1A-CDD8-H`

- Opaqueness – IDs work as a string that you don't need to examine for the Manifest to work correctly. Note that IDs are case insensitive (case should be ignored when matching)
- Uniqueness – Two IDs identifying different objects cannot have the same value (string). The use of <type> with values like 'experienceid', 'cid' and 'audtrackid' ensures that different types of IDs are not confused. The combination <scheme> and <SSID> ensured ID values are unique.
- Base ID extraction – An ID of a given <scheme> can be extracted from the Manifest ID. For example, if the scheme is "eidr-s", the EIDR ID follows the colon. `md:experienceid:eidr-s:`***`5EE7-A973-819A-DC1A-CDD8-H`***. If eidr-x is used, the EIDR is between that colon and the next colon; for example, `md:experienceid:eidr-s:`***`5EE7-A973-819A-DC1A-CDD8-H`***`:trailer.1`. Other ID types can be similarly extracted.

Identifiers should use the "md:" form, preferably using EIDR. Note that in almost all cases the Edit-level EIDR is used. When there is only one offering for a given title, use the

EIDR-S form. For example, md:cid:eidr-s:<edit-level EIDR>. If you can't use EIDR, use the 'org:' form; for example, `md:cid:org:craigsmovies.com:12345ABCDEF`.

2.3.1 Avails ProductID and ALID

This section addresses specifically how to populate ALIDExperienceMap/ALID when using EMA Avails [AVAILS].

If the XML version of EMA Avails is used, the /Avail/ALID is used in this field. In this case, a match is a case-insensitive string match.

If the Excel version of Avails is used, the answer is more complicated~~ as the 'md:' identifier form is not typically used in the spreadsheet.~~. The following table defines

ID Source	**Spreadsheet Column (example)**	**Manifest – per spec (example)**	**Manifest – acceptable (example)**
Cut/Edit level EIDR	ProductID (1489-49A2-3956-4B2D-FE16-5)	EIDR-S form (md:alid:eidr-s:1489-49A2-3956-4B2D-FE16-5)	EIDR (1489-49A2-3956-4B2D-FE16-5)
Studio-proprietary alt cut ID	AltID (VIEW_MT_2006)	ORG form (md:alid:org:craigsmovies.com:VIEW_MT_2006)	craigsmovies.com:VIEW_MT_2006

3 COMMON USE CASES ~~–~~ – FEATURE MANIFEST

This section provides specific instructions on how to construct a Manifest given common use cases. Metadata is discussed in Section 4.

Each of these use cases builds on the previous use case. Associated with each use case is an example Manifest file that corresponds with that use case. The use cases and their associated example files are as follows

Use Case	Example File
Simple Movie with Trailer	http://www.movielabs.com/md/mmc/examples/ManifestCore_Example1_simple.xml
+ Multi-language	http://www.movielabs.com/md/mmc/examples/ManifestCore_Example1_multilang.xml
+ Forced subtitles	http://www.movielabs.com/md/mmc/examples/ManifestCore_Example1_forcedsub.xml
+ Multiple trailers	http://www.movielabs.com/md/mmc/examples/ManifestCore_Example1_trailers.xml
+ Dub cards	http://www.movielabs.com/md/mmc/examples/ManifestCore_Example1_dubcards.xml
+ Pre-roll	http://www.movielabs.com/md/mmc/examples/ManifestCore_Example1_preroll.xml
+ Pre-order	http://www.movielabs.com/md/mmc/examples/ManifestCore_Example1_preorder.xml

Additional examples can be found on Google's help site [GoogleHelp].

3.1 Simple Movie with Trailer

This use case defines the basic structure that is used across all MMC Manifests.

3.1.1 Inventory, Presentation and PictureGroup

The Inventory contains the following (English only)

- Audio track for feature
- Audio track for trailer
- Video track for feature
- Video track for trailer
- Subtitle track for feature
- Subtitle track for trailer (although many trailers do not have subtitle tracks)
- Image references for feature metadata
- Image references for trailer metadata
- Reference to Metadata – this could be separate in some workflows. At a minimum, metadata complies with Media Entertainment Core [MEC].
    - Reference to Feature metadata

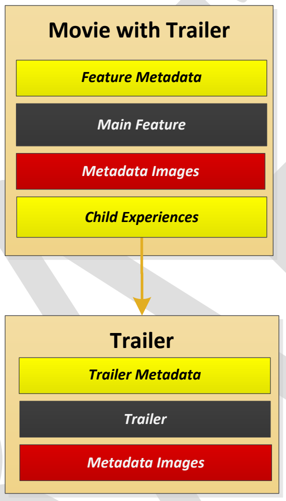
- Reference to Trailer metadata

A Presentation element is created for each set of video, audio and subtitle tracks that play together (conform). Typically, there is a Presentation for the feature and a Presentation for the trailer. Note: all tracks that are designed to be played together should be in the same Presentation (i.e., all language tracks that conform to the video).

One PictureGroup is created for feature metadata images. A second PictureGroup is created for trailer metadata images.

3.1.2 Experience Structure

A simple movie with a trailer has two Experience elements as illustrated here:



The root Experience is constructed as follows

- A unique ExperienceID in the md: format, preferably using the eidr-s or eidr-x format.
- ContentID references metadata for the feature (i.e., matches the ContentID in the Inventory)
- PictureGroupID for metadata images
- An Audivisual instance that references the Presentation for the feature
- An ExperienceChild element referencing the Trailer Experience (ExperienceID is the ID of the trailer Experience). Relationship is 'ispromotionfor'.

The trailer Experience is similar to the root Experience, except that references are to the trailer metadata, Presentation and PictureGroups. Also, the trailer Experience has no ExperienceChild.

3.2 Languages

These use cases cover the addition of languages.

3.2.1 Multi-language movie, no dub cards

Additional languages require the following

- Additional audio and subtitle tracks in Inventory
- Reference to those tracks in Presentation
- Optionally, add LocalizedInfo instances to metadata for new languages (not covered here)

The following illustrates the modifications to Inventory. The additional audio track (shown in ***italic bold***) is added below the other audio track. Audio tracks can be added to the feature video or to the trailer in this manner. Note that dubbed='true' to indicate this track has dubbed audio. Subtitle tracks are added in a similar manner. Note that whether a subtitle track is a 'caption' (Type="SDH") or a 'subtitle' (Type="normal") is determined by the Type field, not the file type.

```
<!--  audio of the main movie  -->
<Audio AudioTrackID="md:audtrackid:eidr-x:AD07-310C-C59D-6785-C63A-G:feature.audio.en">
   <md:Type>primary</md:Type>
   <md:Language>en</md:Language>
   <ContainerReference>
      <ContainerLocation>file://resources/CounselorThe_AD07-310C-C59D-6785-C63A-
_feature_video_ENG.mpg</ContainerLocation>
      </ContainerReference>
</Audio>
<!— addition of French audio track -->
<Audio AudioTrackID="md:audtrackid:eidr-x:AD07-310C-C59D-6785-C63A-G:feature.audio.fr">
   <md:Type>primary</md:Type>
   <md:Language dubbed='true'>fr</md:Language>
   <ContainerReference>
      <ContainerLocation>file://resources/CounselorThe_AD07-310C-C59D-6785-C63A-
G_feature_video_FR.mpg</ContainerLocation>
   </ContainerReference>
</Audio>
```

Subtitle tracks are also added. Note that if subtitles are just for language use (as opposed to accessibility), they are typed as 'normal'

```
<Subtitle SubtitleTrackID="md:subtrackid:eidr-x:AD07-310C-C59D-6785-C63A-G:feature.caption.fr"
   <md:Type>normal</md:Type>
   <md:Language>fr</md:Language>
   <ContainerReference>
      <ContainerLocation>file://resources/CounselorThe_AD07-310C-C59D-6785-C63A-
G_closed_caption_FR.scc</ContainerLocation>
   </ContainerReference>
</Subtitle>
```

Similarly, the tracks must be added to the Presentation as follows. This example shows the addition of French audio and subtitle.

```
<Presentation PresentationID="md:presentationid:eidr-x:AD07-310C-C59D-6785-C63A-
G:feature.presentation">
   <TrackMetadata>
      <TrackSelectionNumber>0</TrackSelectionNumber>
      <VideoTrackReference>
         <VideoTrackID>md:vidtrackid:eidr-x:AD07-310C-C59D-6785-C63A-
G:feature.video</VideoTrackID>
      </VideoTrackReference>
      <AudioTrackReference>
         <AudioTrackID>md:audtrackid:eidr-x:AD07-310C-C59D-6785-C63A-
G:feature.audio.en</AudioTrackID>
      </AudioTrackReference>
      <AudioTrackReference>
         <AudioTrackID>md:audtrackid:eidr-x:AD07-310C-C59D-6785-C63A-
G:feature.audio.fr</AudioTrackID>
      </AudioTrackReference>
      <SubtitleTrackReference>
         <SubtitleTrackID>md:subtrackid:eidr-x:AD07-310C-C59D-6785-C63A-
G:feature.caption.en</SubtitleTrackID>
      </SubtitleTrackReference>
      <SubtitleTrackReference>
         <SubtitleTrackID>md:subtrackid:eidr-x:AD07-310C-C59D-6785-C63A-
G:feature.caption.fr</SubtitleTrackID>
      </SubtitleTrackReference>
   </TrackMetadata>
</Presentation>
```

3.2.2 Multi-language movie with forced subtitles

Forced subtitles are includes in addition to ‘normal’ (language) or ‘SDH’ subtitles. These subtitles are played when other subtitles are not being played.

Forced subtitles are included in

- Inventory. Note that Type=‘forced’.

```
<!--  subtitle and caption components for the feature  -->
<Subtitle SubtitleTrackID="md:subtrackid:eidr-x:AD07-310C-C59D-6785-C63A-G:feature.caption.en">
   <md:Type>SDH</md:Type>
   <md:Language>en</md:Language>
   <ContainerReference>
      <ContainerLocation>file://resources/CounselorThe_AD07-310C-C59D-6785-C63A-
G_closed_caption_ENG.scc</ContainerLocation>
   </ContainerReference>
</Subtitle>
<!-- Forced Subtitle -->
<Subtitle SubtitleTrackID="md:subtrackid:eidr-x:AD07-310C-C59D-6785-C63A-G:feature.forced.en">
   <md:Type>forced</md:Type>
   <md:Language>en</md:Language>
   <ContainerReference>
      <ContainerLocation>file://resources/CounselorThe_AD07-310C-C59D-6785-C63A-
G_closed_forced_ENG.scc</ContainerLocation>
   </ContainerReference>
</Subtitle>
```

- Presentation

```
<SubtitleTrackReference>
   <SubtitleTrackID>md:subtrackid:eidr-x:AD07-310C-C59D-6785-C63A-
G:feature.caption.en</SubtitleTrackID>
</SubtitleTrackReference>
<!-- Forced Subtitle -->
<SubtitleTrackReference>
   <SubtitleTrackID>md:subtrackid:eidr-x:AD07-310C-C59D-6785-C63A-
G:feature.forced.en</SubtitleTrackID>
</SubtitleTrackReference>
```

3.2.3 Multi-language with multiple trailers

Trailers are added to the Manifest by addition the Inventory, Presentation and Experience elements (just like any other audiovisual).

It is important to set certain metadata objects correctly to ensure the correct trailer will be used. Trailer selection is generally performed based on trailer's metadata (BasicMetadata), as follows:

- Select the trailer for correct country based on BasicMetadata/ReleaseHistory
- Inventory/Audio/Language indicates which audio tracks are available. Generally, the player looks for a trailer with a matching audio track. Inventory/Video/Language can also be encoded if the video is language-specific. Note that locale part of the language tag can be used as part of the logic to select the appropriate trailer.
- If there are no matches, look for trailer with Experience/Audiovisual/SubType="Default Trailer"

Following are examples of these metadata objects being set

- ReleaseHistory – only needed for county-specific trailers. A DistrTerritory/country or DistrTerritory/region element specifies where trailer should be played.

```
<md:ReleaseHistory>
   <md:ReleaseType>Theatrical</md:ReleaseType>
   <md:DistrTerritory>
      <md:country>US</md:country>
   </md:DistrTerritory>
   <Date>2013</Date>
</md:ReleaseHistory>
```

- Audio/Language

```
<Audio AudioTrackID="md:audtrackid:eidr-x:AD07-310C-C59D-6785-C63A-G:trailer.2.audio.fr">
   <md:Type>primary</md:Type>
   <md:Language>fr</md:Language>
   <ContainerReference>
      <ContainerLocation>file://resources/CounselorThe_AD07-310C-C59D-6785-C63A-
G_trailer_video_FR.mpg</ContainerLocation>
   </ContainerReference>
</Audio>
```

- Audiovisual/SubType (first one is default and second one is not)

```
<Experience ExperienceID="md:experienceid:eidr-x:AD07-310C-C59D-6785-C63A-G:trailer.1.experience"
version="1.0">
   <ContentID>md:cid:eidr-x:AD07-310C-C59D-6785-C63A-G:trailer.1</ContentID>
   <Audiovisual ContentID="md:cid:eidr-x:AD07-310C-C59D-6785-C63A-G:trailer.1">
      <Type>Promotion</Type>
      <SubType>Default Trailer</SubType>
      <PresentationID>md:presentationid:eidr-x:AD07-310C-C59D-6785-C63A-
G:trailer.1.presentation</PresentationID>
   </Audiovisual>
</Experience>
<!—Alternate Trailer -->
<Experience ExperienceID="md:experienceid:eidr-x:AD07-310C-C59D-6785-C63A-G:trailer.2.experience"
version="1.0">
   <ContentID>md:cid:eidr-x:AD07-310C-C59D-6785-C63A-G:trailer.2</ContentID>
   <Audiovisual ContentID="md:cid:eidr-x:AD07-310C-C59D-6785-C63A-G:trailer.2">
      <Type>Promotion</Type>
      <SubType>Trailer</SubType>
      <PresentationID>md:presentationid:eidr-x:AD07-310C-C59D-6785-C63A-
G:trailer.2.presentation</PresentationID>
   </Audiovisual>
</Experience>
```

3.2.4 Multi-language movie with dub cards

Dub cards are unique segments of video that are associated with an audio dub, typically containing dubbing credits. The method described here provides information that allows a player to select the dub card based on which language is being played. For example, if the user is playing a French audio dub, the French dub card will show after the feature.

Dub cards are facilitated by the PlayableSequence structure. For dub cards, the Playable Sequence consists of the primary Presentation as above followed by a set of dub cards. Note that the Presentation defined in the previous example is unmodified.

The changes required are

- Add Dub cards
- Add PlayableSequence that includes the dub cards
- In Experience, change reference to a Presentation to a PlayableSequence

Dub cards are video clips. Other options exist, but not in the Core definition. They must be included in the Inventory.

```
<!-- video files for the dub cards -->
<Video VideoTrackID="md:vidtrackid:eidr-x:AD07-310C-C59D-6785-C63A-G:dubcard.fr">
   <md:Type>primary</md:Type>
   <md:Picture/>
   <md:CardsetList>
      <md:Cardset>
         <md:Type>DubbingCredit</md:Type>
         <md:Language>fr</md:Language>
      </md:Cardset>
   </md:CardsetList>
   <ContainerReference>
      <ContainerLocation>file://resources/CounselorThe_AD07-310C-C59D-6785-C63A-
G_dubcard_CFR.mpg</ContainerLocation>
   </ContainerReference>
</Video>
<Video VideoTrackID="md:vidtrackid:eidr-x:AD07-310C-C59D-6785-C63A-G:dubcard.pt-br">
   <md:Type>primary</md:Type>
   <md:Picture/>
   <md:CardsetList>
      <md:Cardset>
         <md:Type>DubbingCredite</md:Type>
         <md:Language>pt-BR</md:Language>
      </md:Cardset>
   </md:CardsetList>
   < ContainerReference>
      <ContainerLocation>file://resources/CounselorThe_AD07-310C-C59D-6785-C63A-
G_dubcard_BPO.mpg</ContainerLocation>
   </ContainerReference>
</Video>
```

Dub Cards are Presentations, with only video. For example:

```
<!—Presentations for the dub cards -->
<Presentation PresentationID="md:presentationid:eidr-x:AD07-310C-C59D-6785-C63A-G:dubcard.pt-
br.presentation">
```

```
   <TrackMetadata>
      <TrackSelectionNumber>0</TrackSelectionNumber>
      <!-- only the video track is referenced since it doesn't have an audio track -->
      <VideoTrackReference>
         <VideoTrackID>md:vidtrackid:eidr-x:AD07-310C-C59D-6785-C63A-G:dubcard.pt-
br</VideoTrackID>
      </VideoTrackReference>
   </TrackMetadata>
</Presentation>
<Presentation PresentationID="md:presentationid:eidr-x:AD07-310C-C59D-6785-C63A-G:dubcard.fr-
ca.presentation">
   <TrackMetadata>
      <TrackSelectionNumber>0</TrackSelectionNumber>
      <VideoTrackReference>
         <VideoTrackID>md:vidtrackid:eidr-x:AD07-310C-C59D-6785-C63A-G:dubcard.fr-
ca</VideoTrackID>
      </VideoTrackReference>
   </TrackMetadata>
</Presentation>
```

Note that in this Core definition, only video clips are support (i.e., image clips are not supported).

A Playable Sequence is a mechanism for playing two or more videos in sequence. In this context, we are using a Playable Sequence to play a main video followed by dub cards. It can also be used to play anti-piracy and ratings cards before a main feature.

The Playable Sequence has two sequences, ‘0’ for the main. Note that the main Presentation is always sequence=‘0’. The select the dub card based on the audio being played, set sequence=‘1’ for all dub cards. If a player is not that smart, it can play them all. Following is an example of a Playable Sequence with a main video and two dub cards.

```
<!--Playable Sequences define an order of the feature and dub cards if delivering dub cards -->
<PlayableSequences>
   <PlayableSequence PlayableSequenceID="md:playablesequenceid:eidr-x:AD07-310C-C59D-6785-C63A-
G:feature">
      <Clip sequence="0">
         <PresentationID>md:presentationid:eidr-x:AD07-310C-C59D-6785-C63A-
G:feature.presentation</PresentationID>
      </Clip>
      <Clip sequence="1">
         <PresentationID>md:presentationid:eidr-x:AD07-310C-C59D-6785-C63A-
G:dubcard.fr.presentation</PresentationID>
      </Clip>
      <Clip sequence="1">
         <PresentationID>md:presentationid:eidr-x:AD07-310C-C59D-6785-C63A-G:dubcard.pt-
br.presentation</PresentationID>
      </Clip>
   </PlayableSequence>
</PlayableSequences>
```

3.2.5 Multi-language movie with ratings pre-roll and dub card post-roll

This is included as a forward-looking feature.

Pre-roll (plays before feature) is like post-roll (plays after feature) except that the sequence numbers in PlayableSequence are negative.

This example shows a PlayableSequence that includes a pre-roll of Canadian (CHVRS) ratings.

```
<PlayableSequence PlayableSequenceID="md:playablesequenceid:eidr-x:AD07-310C-C59D-6785-C63A-G:feature.ca">
   <!-- Preroll -->
   <Clip sequence="-1">
   <PresentationID>md:presentationid:URI:http://www.movielabs.com/md/ratings/CA/CHVRS/1/14A</PresentationID>
   </Clip>
   <Clip sequence="0">
      <PresentationID>md:presentationid:eidr-x:AD07-310C-C59D-6785-C63A-G:feature.presentation</PresentationID>
   </Clip>
   <!-- Dubcards -->
   <Clip sequence="1">
      <PresentationID>md:presentationid:eidr-x:AD07-310C-C59D-6785-C63A-G:dubcard.fr.presentation</PresentationID>
   </Clip>
   <!-- Dubcards -->
   <Clip sequence="1">
      <PresentationID>md:presentationid:eidr-x:AD07-310C-C59D-6785-C63A-G:dubcard.pt-br.presentation</PresentationID>
   </Clip>
</PlayableSequence>
```

The sample file has additional changes to illustrate the use of pre-roll across multiple regions in the example, there are Experience instances localized to Canada and not-Canada (i.e., everywhere else).

```
<!-- Root Experience: Everywhere except Canada -->
<Experience ExperienceID="md:experienceid:eidr-x:AD07-310C-C59D-6785-C63A-G:experience.us" version="1.0">
   <!—Note the use of "Excluded" Region -->
   <ExcludedRegion>
      <md:country>CA</md:country>
   </ExcludedRegion>
...
   </Experience>
<!-- Root Experience: Canada -->
<Experience ExperienceID="md:experienceid:eidr-x:AD07-310C-C59D-6785-C63A-G:experience.ca" version="1.0">
   <Region>
      <md:country>CA</md:country>
   </Region>
...
</Experience>
```

These Experiences are referenced in the ALIDExperienceMap:

```
         <ALIDExperienceMap>
         <ALID>md:alid:eidr-s:AD07-310C-C59D-6785-C63A-G</ALID>
         <!-- Preroll (options) -->
         <ExperienceID>md:experienceid:eidr-x:AD07-310C-C59D-6785-C63A-G:experience.us</ExperienceID>
         <ExperienceID>md:experienceid:eidr-x:AD07-310C-C59D-6785-C63A-G:experience.ca</ExperienceID>
      </ALIDExperienceMap>
```

3.3 Pre-order

The MMC includes an *optional* feature to address pre-order conditions. In pre-order windows, the trailers will be available but not the main feature. To address this, create distinct Experience elements for each state. Then use the @condition attribute in ALIDExperienceMap to reference the appropriate Experience. From [Manifest], conditions are as follows:

- ‘Pre-order’ – Asset can be sold, but not fulfilled. A pre-order condition might have a different trailer than the for-sale experience (e.g., ending with “Available on December 15”)
- ‘For-sale’ – (default) Asset can be sold and fulfilled. This is the default and is assumed if @condition is absent.

When an asset, particularly the main feature, is not available in pre-order, PresentationID or PlayableSequenceID is excluded in the main Audiovisual. Metadata should be provided.

The following shows an example of a pre-order Experience. The Audiovisual does not reference a Presentation or Playable Sequence. Trailers are referenced as usual because they are part of what is shown to the consumer in a pre-order offer.

```
<Experience ExperienceID="md:experienceid:eidr-x:AD07-310C-C59D-6785-C63A-G:experience.preorder"
version="1.0">
   <ContentID>md:cid:eidr-s:AD07-310C-C59D-6785-C63A-G</ContentID>
   <Audiovisual ContentID="md:cid:eidr-s:AD07-310C-C59D-6785-C63A-G">
      <Type>Main</Type>
      <SubType>Feature</SubType>
      <!-- No Presentation or PlayableSequence is present.  This indicates this asset is to be
shown, but not played. -->
   </Audiovisual>
   <ExperienceChild>
      <Relationship>ispromotionfor</Relationship>
      <ExperienceID>md:experienceid:eidr-x:AD07-310C-C59D-6785-C63A-
G:trailer.1.experience</ExperienceID>
   </ExperienceChild>
   <!-- MultiLanguage -->
   <ExperienceChild>
      <Relationship>ispromotionfor</Relationship>
      <ExperienceID>md:experienceid:eidr-x:AD07-310C-C59D-6785-C63A-
G:trailer.2.experience</ExperienceID>
   </ExperienceChild>
</Experience>
```

The “Pre-order” conditions refers the pre-order Experience:

```
<ALIDExperienceMaps>
   <ALIDExperienceMap>
   <ALID>md:alid:eidr-s:AD07-310C-C59D-6785-C63A-G</ALID>
   <ExperienceID>md:experienceid:eidr-x:AD07-310C-C59D-6785-C63A-G:experience.us</ExperienceID>
   <!-- Locale -->
   <ExperienceID>md:experienceid:eidr-x:AD07-310C-C59D-6785-C63A-G:experience.ca</ExperienceID>
   <!-- Preorder -->
   <ExperienceID condition="Pre-order">md:experienceid:eidr-x:AD07-310C-C59D-6785-C63A-
G:experience.preorder</ExperienceID>
   </ALIDExperienceMap>
</ALIDExperienceMaps>
```

Note that this should not affect other elements (Inventory, Presentation, etc.) unless there are assets that are only available prior to acquisition.

4 COMMON USE CASES: METADATA

This section provides specific instructions on how to construct Basic Metadata given common use cases. Manifest structure is discussed in Section 3.

Examples are provided for use cases.

Feature Use Case	Example File
Basic Metadata – Movie (simplified)	http://www.movielabs.com/md/mmc/examples/ManifestCore-Example1_MEC-movie-simple.xml
Basic Metadata - Movie	http://www.movielabs.com/md/mmc/examples/ManifestCore-Example1_MEC-movie.xml
Basic Metadata - Trailer	http://www.movielabs.com/md/mmc/examples/ManifestCore_Example1_MEC-Trailer1.xml http://www.movielabs.com/md/mmc/examples/ManifestCore_Example1_MEC-Trailer2.xml

TV Use Case	Example File
Basic Metadata – Series	http://www.movielabs.com/md/mmc/examples-draft/VEEP_Series_mec.xml
Basic Metadata - Season	http://www.movielabs.com/md/mmc/examples-draft/VEEP_Season5_mec.xml
Basic Metadata - Episode	http://www.movielabs.com/md/mmc/examples-draft/VEEP_Season5_E5_mec.xml

Metadata may be included in the Manifest or delivered in separate files.

4.1 Basic Metadata

A recommended set of Basic metadata can be found in Media Entertainment Core [MEC] Section, 2.2.1. Following is the subset of MEC that is required for the Media Manifest Core.

4.1.1 Feature and Bonus Metadata

Element or Attribute		Usage Rules
BasicMetadata-type		Required
	@ContentID	Required
	UpdateNum	Shall be included if the record is an update (i.e., not the first record distributed)
	LocalizedInfo	At least one instance required

		@language	Required
		default*	must be included for one instance of LocalizedInfo for the language of original production
		TitleDisplayUnlimited	Required
		OriginalTitle	Required
		Summary4000	Required. (2000 preferred)
		Genre	Exactly one primary genre shall be included. It will be from http://www.movielabs.com/md/mec/mec_primary_genre.html @source='http://www.movielabs.com/md/mec/mec_primary_genre.html'. @level='0'. Any additional genres may be included.
		ArtReference	At least one instance is mandatory, additional instances are optional
	ReleaseYear		Required
	ReleaseHistory		Only required for Trailers when DistrTerritory is needed to specify locale of trailer.
	WorkType		Required
	RatingSet		SHALL be included for all available ratings in the regions where Retailers are authorized to sell this content. All elements and attributes should be included if applicable to the rating. The condition attribute should be used if the primary purpose of the edit is a derivation from a parent for the purposes of ratings change (e.g., airline edit or 'unrated edition').

4.1.2 Trailer Metadata

Trailer metadata is generally the same as metadata for any other object. Following is some additional guidance:

- If there is no trailer artwork, ArtReference can reference the feature's artwork
- If the trailer is specific to one or more specific countries ReleaseHistory/DistrTerritory is included for each country.
- Trailers should be rated in accordance with the rating rules for trailers in given regions. Note that in some countries, ~~that~~the trailer's rating is the same as the movie. In others, there are special ratings (e.g., MPAA "Green Band" and "Red Band").
- The Parent should be included with @relationshipType='ispromotionfor' and ParentContentID set to the ContentID of the main feature.

4.2 Packaging Metadata

4.2.1 Metadata in standalone file

Metadata can be in its own file. This file is constructed in accordance with the Media Entertainment Core [MEC]. The constraints defined in Section 4.1 can be used, although a full MEC implementation is preferred.

The external metadata file is referenced by BasicMetadata@ContentID. That is, BasicMetadata@ContentID matches a ContentID from the Media Manifest. For example, the following from a Media Manifest

```
<Experience ExperienceID="md:experienceid:eidr-x:AD07-310C-C59D-6785-C63A-G:experience.ca"
version="1.0">
   <ContentID>md:cid:eidr-s:AD07-310C-C59D-6785-C63A-G</ContentID>
   <Audiovisual ContentID="md:cid:eidr-s:AD07-310C-C59D-6785-C63A-G">
      <Type>Main</Type>
      <SubType>Feature</SubType>
   …
```

~~Matches~~matches this from the metadata file

```
<mdmec:Basic ContentID="md:cid:eidr-s:AD07-310C-C59D-6785-C63A-G">
```

The Media Manifest has the mechanism for referencing images. Therefore, image references within BasicMetadata (i.e., LocalizedInfo/ArtReference) must contain an Image ID. For example,

```
<ArtReference resolution="1789x2560">imageid:eidr-x:AD07-310C-C59D-6785-C63A-
G:art.en</ArtReference>
```

4.2.2 Metadata in Manifest

The Manifest should indicate where the metadata file can be located. This is done by including Inventory/Metadata/ContainerReference. The contents of ContainerReference depends on how the metadata file will be delivered. Following is an example:

```
<Metadata ContentID="md:cid:eidr-s:AD07-310C-C59D-6785-C63A-G">
   <ContainerReference type="common">
      <ContainerLocation>file://resources/ManifestCore-Example1-MEC-movie.xml</ContainerLocation>
   </ContainerReference>
</Metadata>
```

5 COMMON USE CASE – TV MANIFEST

MMC for TV is essentially a superset of MMC for movies, with the additional information being series and season information.

Use Case	Example Files
Series	http://www.movielabs.com/md/mmc/examples-draft/VEEP_Series_manifest.xml
Season	http://www.movielabs.com/md/mmc/examples-draft/VEEP_Season5_manifest.xml
Episode	http://www.movielabs.com/md/mmc/examples-draft/VEEP_Season5_E5_manifest.xml
Episode with Bonus	http://www.movielabs.com/md/mmc/examples-draft/VEEP_Season5_E5_WithBonus_manifest.xml
Season with episode reordering	http://www.movielabs.com/md/mmc/examples-draft/VEEP_Season5_manifest_WithReordering.xml

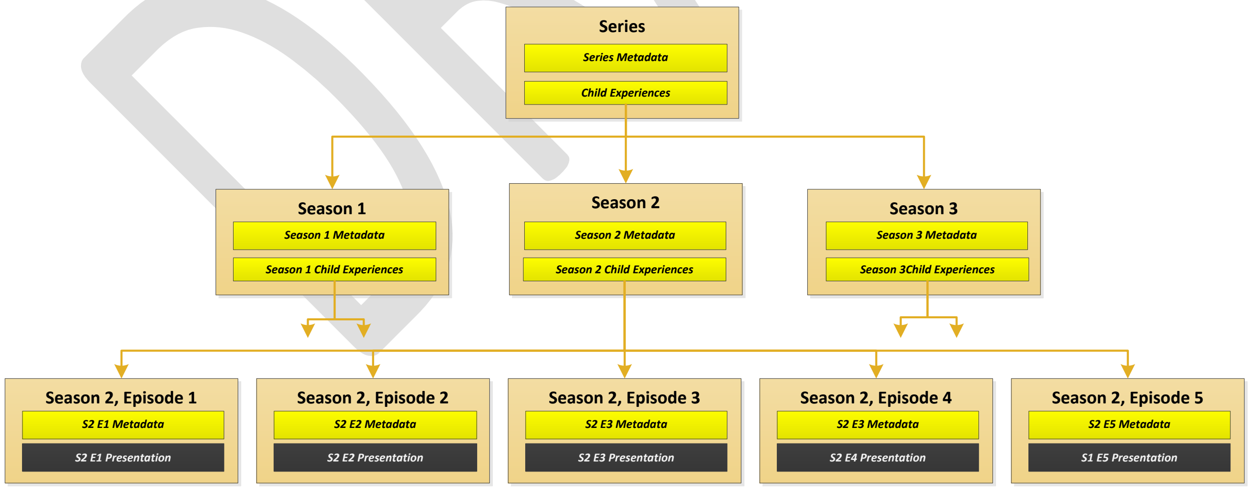
5.1 Summary of Approach

To support common workflows between movies and TV, MMC delivers TV episodes much like movies (features). There are differences in the metadata, but otherwise the structure is the same. Bonus material associated with the episode and promotional material (e.g., teasers) are structured the same as with movies.

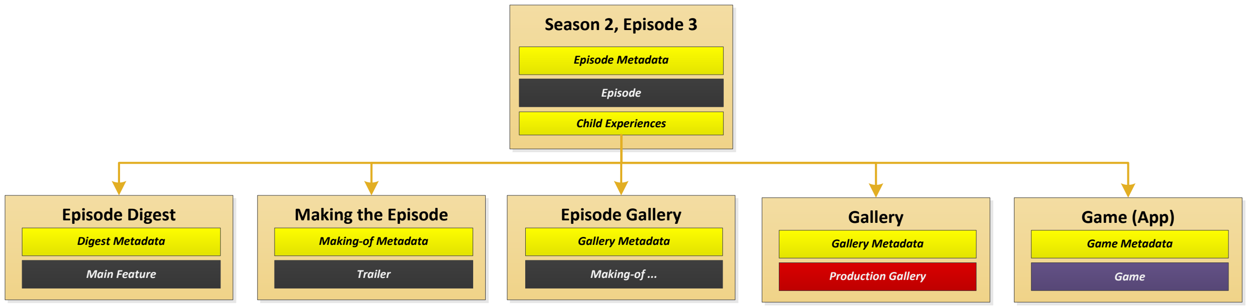
Seasons and series are delivered as additional manifests.

5.2 Content Structure

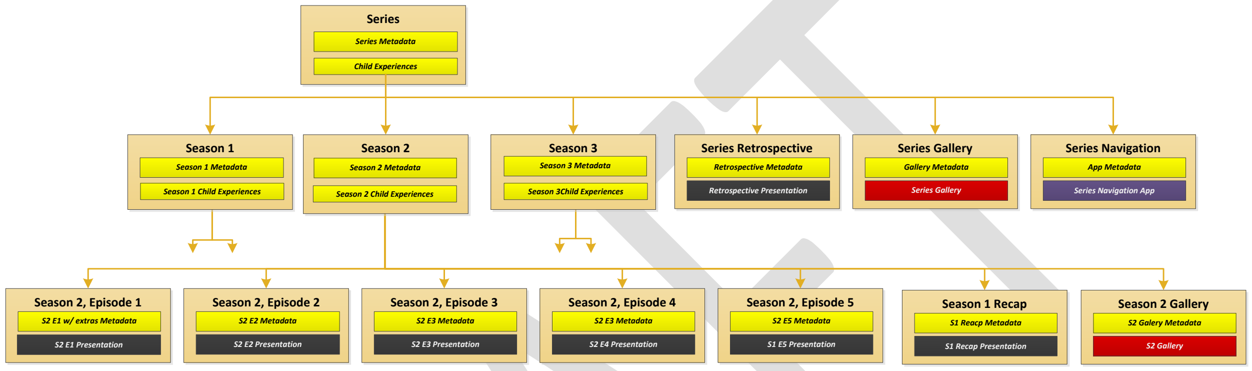
The following illustration shows a generic TV series with Season 2 detailed.



When episodes have additional material (e.g., bonus material), it would look something like this:



Series and seasons can have their own bonus, looking something like this:



5.2.1 Connecting Season to Series via ALID

The Avails spec [Avails] does not currently define a Series offering which means there is no series ALID. To find the Series parent object, ALIDExperienceMap/RelatedExperienceID is used.

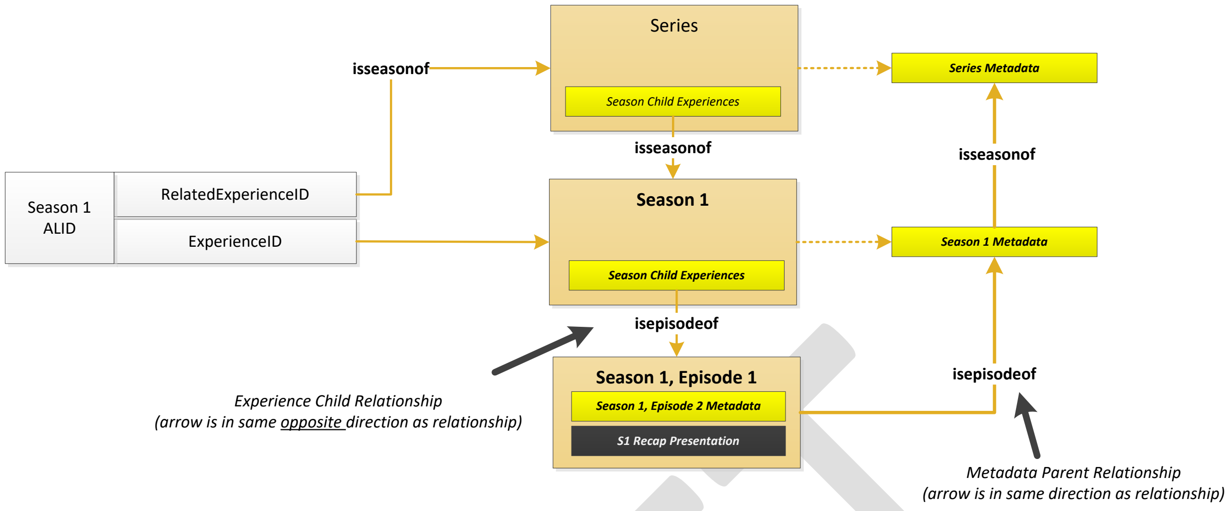
For every ALIDExperienceMap instance containing Season ALIDs, there should be a RelatedExperienceID instance for each Series Experience.

RelatedExperienceID is encoded as follows:

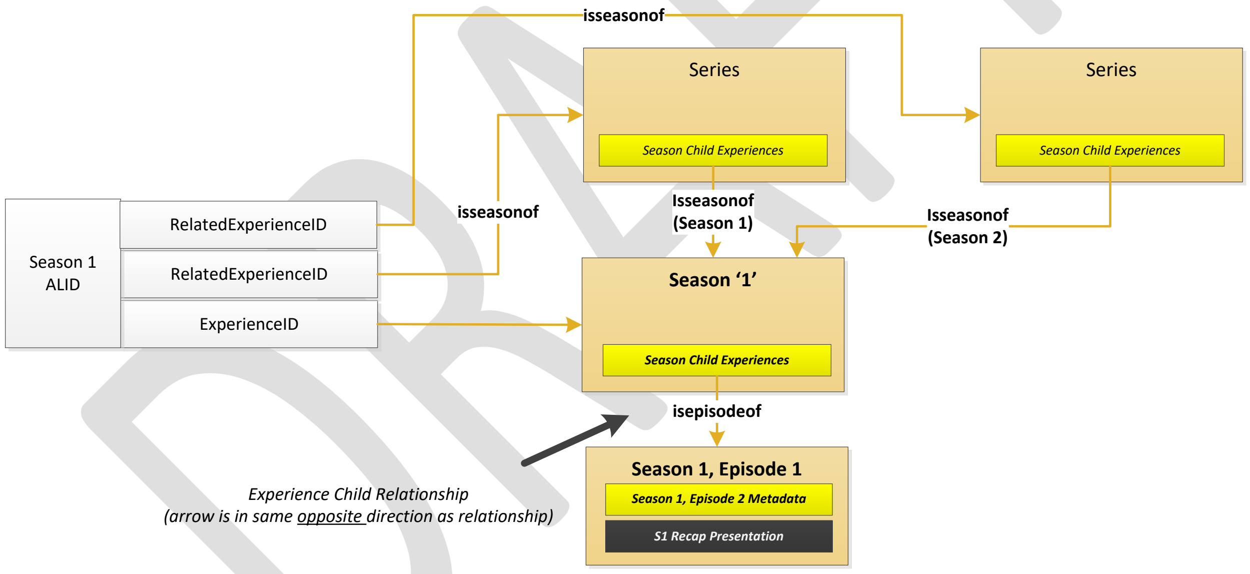
- Relationship = 'isseasonof'
- SequenceInfo should not be used. This should be derived from ExperienceChild within the Series Experience referenced by ExperienceID.
- ExperienceID = the ID of the Experience containing the Series information.
- ExternalManifestID should contain the ManifestID of the Manifest with the Series in it. Note that this should be known since the Series manifest must always be present when a Season manifest is delivered.

Note that unless there's an ALID defining the Series (i.e., the Series is Availed), there need not be an ALIDExperienceMap in the Series Manifest.

Typically, it looks like this:



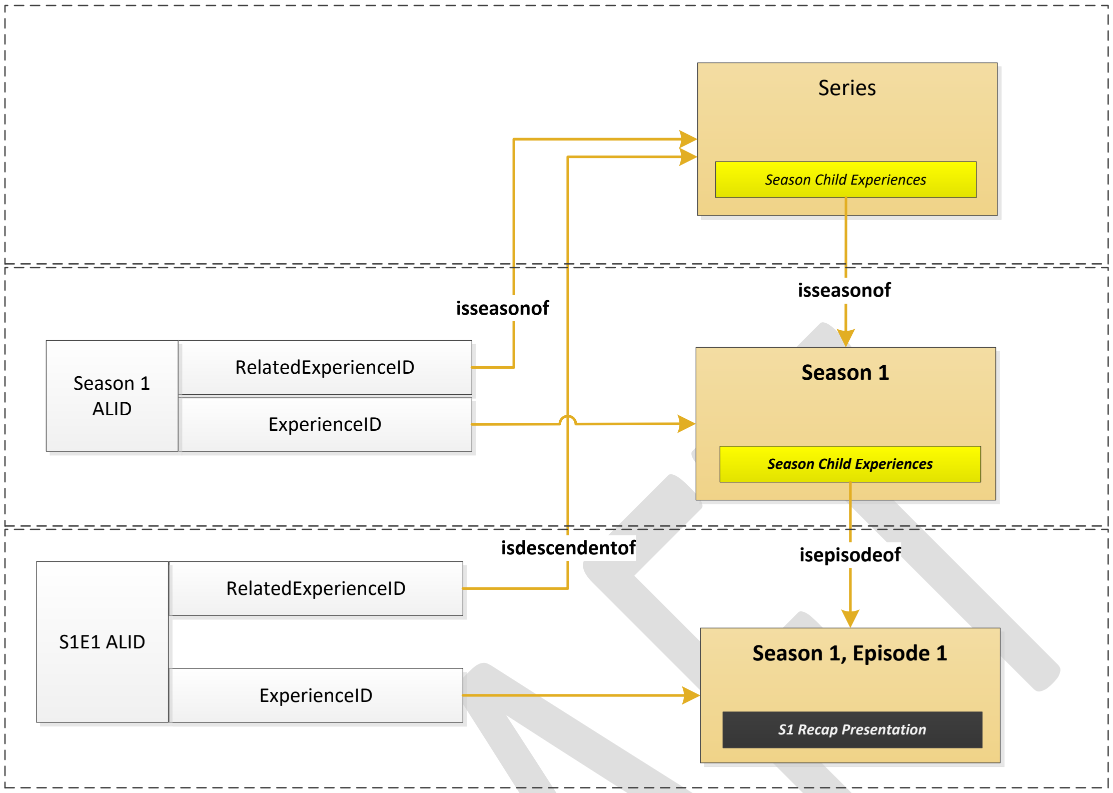
If there are multiple Series Experiences that are regionalized (i.e., reordered seasons or different number of seasons), there should be a RelatedExperienceID instance for each Series Experience. All Series Experiences would be referenced, as shown below:



5.2.2 Finding Root Parent Experience

There should always be a RelatedExperience that references the highest level object. This allows the recipient to build an Experience tree starting from the top. This provides full context for the object.

The following illustrates three Manifests (surrounded by dashed boxes), one for series, one for a season and one for an episode. Note that both the season and episode reference the Series (i.e., the highest-level object). Note that the relationship between an episode and a series is 'isdependentof'.



The following examples show ALIDExperienceMaps for Season and Series

Season:

```
<manifest:ALIDExperienceMap>
   <manifest:ALID>md:alid:eidr-x:2D99-3C1C-9F31-3E10-3411-1:de.seasonpass</manifest:ALID>
   <manifest:ExperienceID>md:experienceid:eidr-x:2D99-3C1C-9F31-3E10-3411-
1:de</manifest:ExperienceID>
   <manifest:RelatedExperienceID>
      <manifest:Relationship>isseasonof</manifest:Relationship>
      <manifest:ExperienceID>md:experience:org:number.hbo.com:787232.1</manifest:ExperienceID>
      <manifest:ExternalManifestID>md:manifestid:eidr-s:CF5A-AB7E-A4DB-35FA-BAC5-
M</manifest:ExternalManifestID>
   </manifest:RelatedExperienceID>
</manifest:ALIDExperienceMap>
```

Episode:

```
<manifest:ALIDExperienceMap>
   <manifest:ALID>md:alid:org:number.hbo.com:787232</manifest:ALID>
   <manifest:ExperienceID>md:experience:org:number.hbo.com:787232.1</manifest:ExperienceID>
   <manifest:RelatedExperienceID>
      <manifest:Relationship>isdescendentof</manifest:Relationship>
      <manifest:ExperienceID>md:experience:org:number.hbo.com:787232.1</manifest:ExperienceID>
      <manifest:ExternalManifestID>md:manifestid:eidr-s:CF5A-AB7E-A4DB-35FA-BAC5-
M</manifest:ExternalManifestID>
   </manifest:RelatedExperienceID>
</manifest:ALIDExperienceMap>
```

5.3 Workflows

There are three factors that drive the sequence of delivery and determine what is delivered. The first and most important is whether the season has already aired, or whether episodes are still being aired. The second is whether there is bonus material attached to seasons or series. This is a complexity because there is no analogy in MMC for features. The third is whether there is alternate episode ordering or different episodes for different territories. It turns out the solutions for the second and third factors result in the same workflow, so they are lumped together.

These conditions yield four use cases addressed here:

- Case 1: No season/series bonus
  - Case 1a: Complete Season
  - Case 1b: Active Season
- Case 2: With Season/Series Bonus or alternate episode ordering
  - Case 2a: Complete Season
  - Case 2b: Active Season

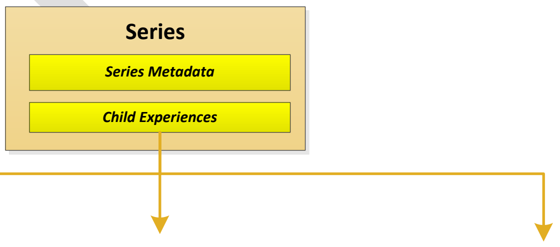
5.3.1 Complete Season

A complete season is the simplest use case as all information is known upfront. There is no expectation that future updates are expected (corrections notwithstanding).

Regardless of all information being known, separate Series, Season and Episode manifests are delivered. This is to increase commonality with other TV and movie workflows.

5.3.1.1 Series Manifest

The Series manifest is delivered once. It includes an ExperienceChild for referencing each Season (via ExperienceID).

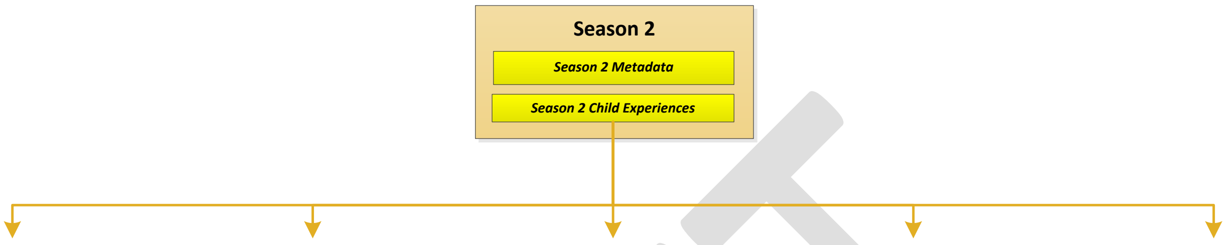


```
<!-- Season 1 -->
<manifest:ExperienceChild>
   <manifest:Relationship>isseasonof</manifest:Relationship>
   <manifest:ExperienceID>md:experienceid:eidr-s:5E1E-908E-5C95-1308-3D18-
S</manifest:ExperienceID>
   <manifest:ExternalManifestID>md:manifestid:eidr-s:5E1E-908E-5C95-1308-3D18-
S</manifest:ExternalManifestID>
</manifest:ExperienceChild>
```

5.3.1.2 Season Manifests

Each season manifest is delivered individually. That is, if there are three seasons in the series, there are three season manifests delivered.

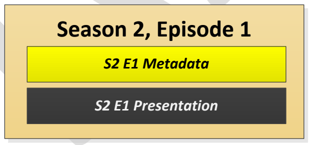
Each season manifest includes an ExperienceChild for referencing each episode (via ExperienceID).



```
<!-- Episode 1 -->
<manifest:ExperienceChild>
   <manifest:Relationship>isepisodeof</manifest:Relationship>
   <manifest:ExperienceID>md:experienceid:org:number.hbo.com:78728</manifest:ExperienceID>
   <manifest:ExternalManifestID>md:manifestid:org:number.hbo.com:78728</manifest:ExternalManifest
ID>
</manifest:ExperienceChild>
<!-- Episode 2 -->
<manifest:ExperienceChild>
   <manifest:Relationship>isepisodeof</manifest:Relationship>
   <manifest:ExperienceID>md:experienceid:org:number.hbo.com:78729</manifest:ExperienceID>
   <manifest:ExternalManifestID>md:manifestid:org:number.hbo.com:78729</manifest:ExternalManifest
ID>
</manifest:ExperienceChild>
```

5.3.1.3 Episode Manifests

Each episode is delivered individually. That is, if there are ten episodes in each of three seasons, thirty episode manifests are delivered. Each episode manifest contains a single episode.



```
<manifest:Experience ExperienceID="md:experience:org:number.hbo.com:787232.1" version="1.0">
   <manifest:Language>de</manifest:Language>
   <manifest:ContentID>md:cid:org:number.hbo.com:787232:</manifest:ContentID>
   <manifest:Audiovisual ContentID="md:cid:org:number.hbo.com:787232:">
      <manifest:Type>Main</manifest:Type>
   <manifest:PresentationID>md:presentationid:org:number.hbo.com:787232:main.us</manifest:Present
ationID>
   </manifest:Audiovisual>
</manifest:Experience>
```

5.3.2 Active Season

5.3.2.1 Series Manifest

The Series Manifest is sent once at the beginning of each new season. It includes the new season. Also, the within the updated Manifest Manifest/updateNum is incremented to reflect the manifest update.

Note that the Series Manifest is the same that delivered for a a completed season with the exception that it includes the season about to be aired (or is currently being aired).

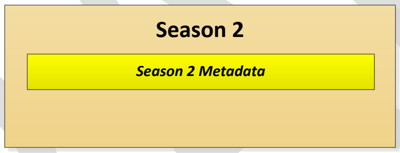
5.3.2.2 Season Manifest

The Season Manifest is sent to reflect the episodes that are being included.

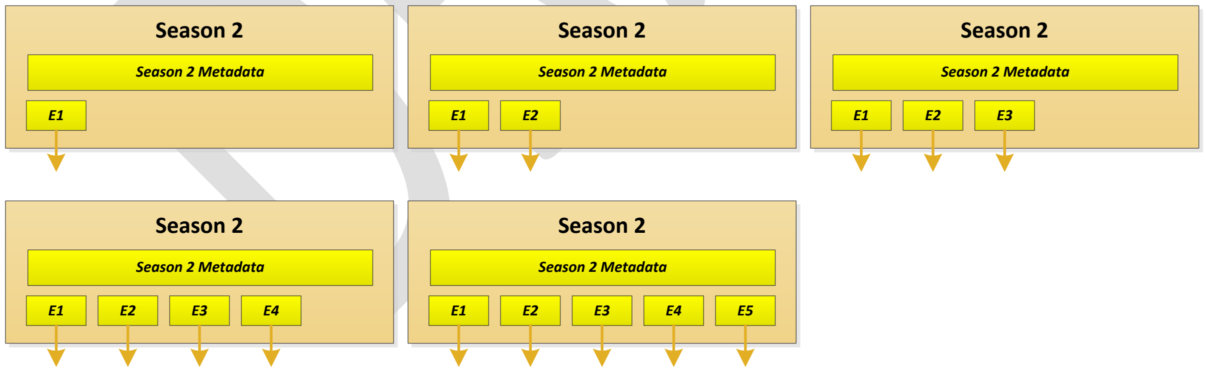
Initially, the Season Manifest can be sent with no episodes. That is, there the Season Experience has not child Experiences.

As Episodes become available, they are included in the Season Manifest. That is, an update of the Manifest is sent for every Episode update. However no updates to the Series Manifest are required.

The Season Manifest can be delivered without any episodes. This allows the retailer to create a 'container' for the upcoming information.



Subsequent Season Manifests are updated and delivered as each episode becomes available. The Season Manifests reference all released episodes (i.e., cumulatively)



```
<!—Inventory contains season metadata images -->
   <manifest:Inventory>
      <!-- MEC images -->
      <manifest:Image ImageID="md:imageid:eidr-x:2D99-3C1C-9F31-3E10-3411-1:season5-photo.de">
      …
      </manifest:Image>
      <manifest:Image ImageID="md:imageid:eidr-x:CF5A-AB7E-A4DB-35FA-BAC5-M:season5-photo.en">
      …
      </manifest:Image>
   </manifest:Inventory>
```

```
    <manifest:Experiences>
        <manifest:Experience ExperienceID="md:experienceid:eidr-x:2D99-3C1C-9F31-3E10-3411-1:de"
version="1.0">
            <manifest:Language>de-DE</manifest:Language>
            <manifest:ContentID>md:cid:eidr-s:CF5A-AB7E-A4DB-35FA-BAC5-M</manifest:ContentID>
<!—Initial delivery is empty.  Subsequent deliveries will have episodes as ExperienceChild -->

        </manifest:Experience>
    </manifest:Experiences>
    <manifest:ALIDExperienceMaps>
        <manifest:ALIDExperienceMap>
            <manifest:ALID>md:alid:eidr-x:2D99-3C1C-9F31-3E10-3411-1:de.seasonpass</manifest:ALID>
            <manifest:ExperienceID>md:experienceid:eidr-x:2D99-3C1C-9F31-3E10-3411-
1:de</manifest:ExperienceID>
        </manifest:ALIDExperienceMap>
    </manifest:ALIDExperienceMaps>
</manifest:MediaManifest>
```

When episodes are added, the Season Manifest is updated with additional ExperienceChild elements for each like this:

```
<!-- Episode 1 -->
<manifest:ExperienceChild>
    <manifest:Relationship>isepisodeof</manifest:Relationship>
    <manifest:ExperienceID>md:experienceid:org:number.hbo.com:78728</manifest:ExperienceID>
    <manifest:ExternalManifestID>md:manifestid:org:number.hbo.com:78728</manifest:ExternalManifest
ID>
</manifest:ExperienceChild>
<!-- Episode 2 -->
<manifest:ExperienceChild>
    <manifest:Relationship>isepisodeof</manifest:Relationship>
    <manifest:ExperienceID>md:experienceid:org:number.hbo.com:78729</manifest:ExperienceID>
    <manifest:ExternalManifestID>md:manifestid:org:number.hbo.com:78729</manifest:ExternalManifest
ID>
</manifest:ExperienceChild>
```

5.3.2.3 Episode Manifests

Episode Manifests are identical to those delivered in the completed season case. See Section 5.3.1.3.

5.3.3 Complete Season with bonus and/or reordered episodes

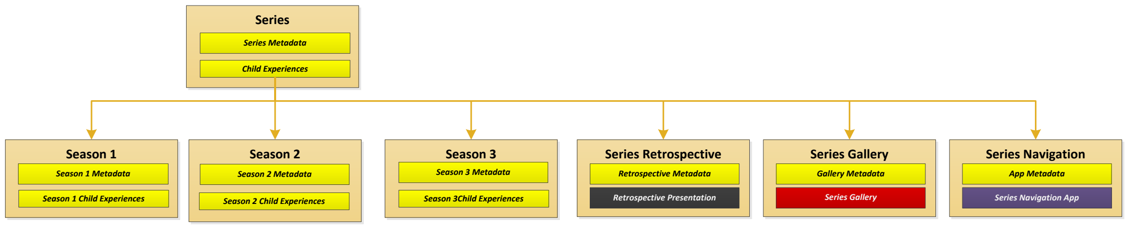
This section defines how to add bonus material to a season, series or episode. It also describes how to offer different episodes and episode ordering in different territories.

Bonus material can be added to the series, season and/or episode. This additional material is included in the respective Manifest as described in this section.

Some series offer different content in different territories. Differences can include episode ordering, different episodes, and different episode edits.

5.3.3.1 Series Manifest with Bonus

Bonus material can be added to a Series by adding children to the Season Experience. The following illustration shows a series with three seasons and three bonus objects.

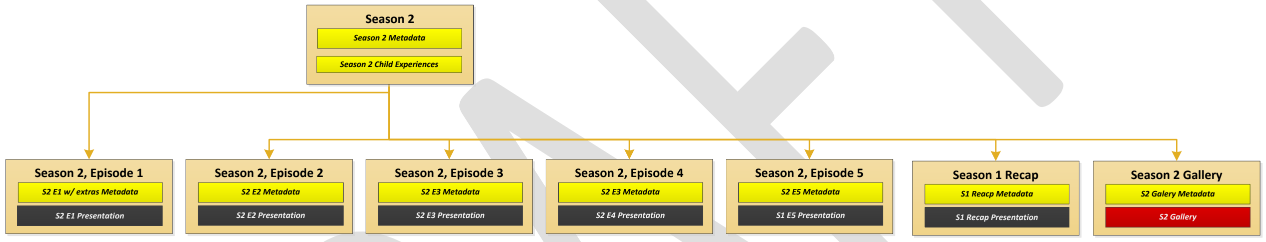


Note that Seasons can be reordered by creating territory-specific Experiences, each ordering the child seasons differently. However, this is rarely done and is not part of the Core.

5.3.3.2 Season Manifest with Bonus

Adding bonus material to a Season requires the addition of ExperienceChild elements to the Season Experience; one Child per bonus.

The following example shows a season with five episodes and two bonus objects.



Each episode and bonus object has its own Experience.

5.3.3.3 Season Manifest with multiple episode Experiences

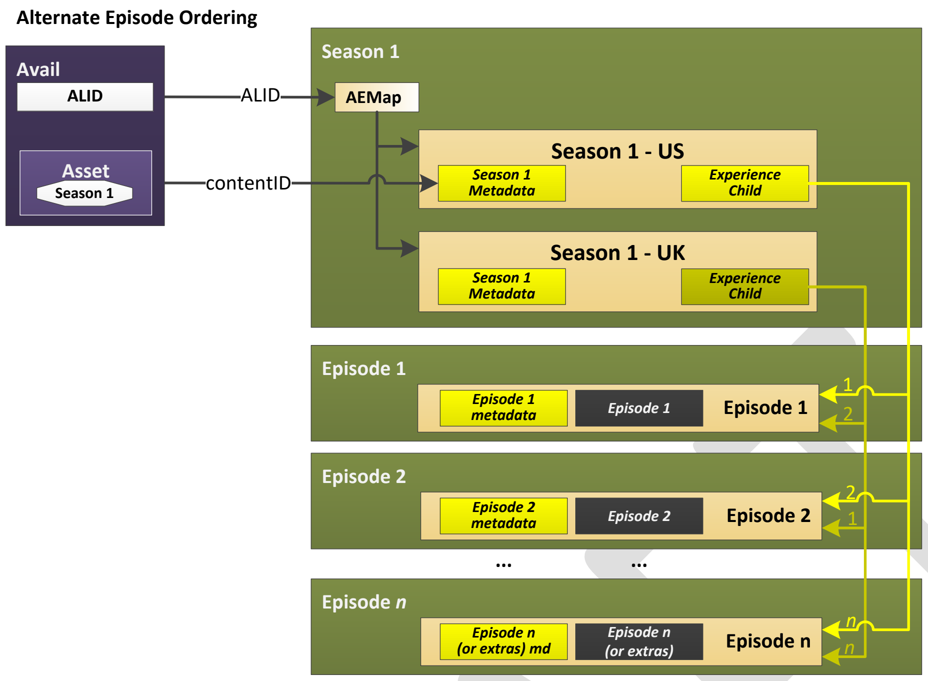
When different territories have different episode order, different episode edits or different episodes (i.e., episodes missing or added), there must be an Experience for each unique combination.

Each of these Experience elements must

- Include the appropriate territory information in Experience/Region or Experience/ExcludedRegion
- Include ExperienceChild, referencing the appropriate child episodes for the territory or territories. That is, ExperienceChild/ExperienceID associated with the correct ExperienceChild/SequenceInfo/Number for that territory

Reference to the appropriate metadata via Experience/ContentID. If metadata is shared between territories, those territories reference the same ContentID.

Episode reordering looks like this:



If multiple Season Experience are included in a single Manifest, then each must be referenced within ALIDExperienceMap.The following fragments of an Experience show two child episodes.

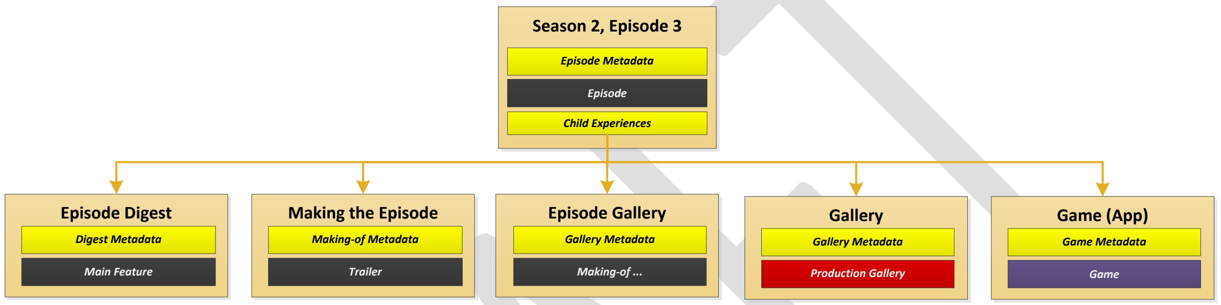
```
      <!-- Episode 1 -->
<manifest:ExperienceChild>
   <manifest:Relationship>isepisodeof</manifest:Relationship>
   <manifest:SequenceInfo>
      <md:Number>1</md:Number>
   </manifest:SequenceInfo>
   <manifest:ExperienceID>md:experienceid:org:number.hbo.com:78728</manifest:ExperienceID>
</manifest:ExperienceChild>
<!-- Episode 2 -->
<manifest:ExperienceChild>
   <manifest:Relationship>isepisodeof</manifest:Relationship>
   <manifest:SequenceInfo>
      <md:Number>2</md:Number>
   </manifest:SequenceInfo>
   <manifest:ExperienceID>md:experienceid:org:number.hbo.com:78729</manifest:ExperienceID>
</manifest:ExperienceChild>
```

This second example reverses episodes 1 and 2. Note that the episodes referred to by the child ExperienceID are reversed (78728 and 78729).

```
<!-- Episode 1 (was episode 2) -->
<manifest:ExperienceChild>
   <manifest:Relationship>isepisodeof</manifest:Relationship>
   <manifest:SequenceInfo>
      <md:Number>1</md:Number>
   </manifest:SequenceInfo>
   <manifest:ExperienceID>md:experienceid:org:number.hbo.com:78729</manifest:ExperienceID>
</manifest:ExperienceChild>
<!-- Episode 2 (was episode 1)-->
<manifest:ExperienceChild>
   <manifest:Relationship>isepisodeof</manifest:Relationship>
   <manifest:SequenceInfo>
      <md:Number>2</md:Number>
   </manifest:SequenceInfo>
   <manifest:ExperienceID>md:experienceid:org:number.hbo.com:78728</manifest:ExperienceID>
</manifest:ExperienceChild>
```

5.3.3.4 Episode Manifest with Bonus

Manifests for Episodes with bonus material are constructed the same as manifests for movies with bonus. That is, the root Experience includes ExperienceChild instances for bonus material, Experience instances are included for each bonus object, and all other necessary information is included (i.e., Presentations, PictureGroups, Inventory, etc.).



5.3.4 Active Season with bonus and/or reordered episodes

5.3.4.1 Series Manifest

Series Manifests are identical to the completed season case.

5.3.4.2 Season Manifest

The Season Manifest is the same as a completed season, but only contains the episodes that are known.

Placeholder Experiences for incomplete episodes, where data is not known, should not be included.

5.3.4.3 Episode Manifests

Episode Manifests are identical to the completed season case.

5.4 Inferring Season and Series

Series and Season Manifests provide structure, provide for the inclusion of bonus material, and allow territory-specific episodes. However, in the simplest cases, there is sufficient information in metadata to infer the season and series Manifests. This information is found in both metadata and in Avails.

For an episode, the MEC Parent object reference the season, and the MEC SequenceInfo/Number object identifies the episode number within that season. For a season, the

Parent object references the series, and the SequenceInfo/Number identifies the order of the season. From these objects, missing season and series Manifests can be inferred.

It is not recommended that the retailer rely on Avails for this information, but it comes early and can help facilitate the workflow. Avails contain Season and Series identifiers. If a retailer encounters an unrecognized identifier, it can infer that the season or series exists. The season's number can be determined from SeasonNumber and episode number can be determined from EpisodeNumber. These values are present in both Excel and XML. Season and Series information can be inferred. Note that while these data in the Avail provide structure, they do not include sufficient metadata. MEC objects for seasons, series and episodes must still be provided (referenced by the various Content ID objects in the Avail).

It should be noted that information is provided in several places: Avail, MEC and Manifest. This not only provides some workflow flexibility; it creates opportunities for QC checks. Inconsistencies in these objects is a strong signal for human check.