Image Delivery
Options for image file delivery

This practice defines various options for delivering metadata images, beyond what is described in Media Manifest Core (MMC). This practice can also be used for delivery of other images (e.g., image-based cards).

REVISION HISTORY

Version	Date	Description
1.0	December 8, 2020	Initial publication

1 Approach defined in MMC

This document assumes a working understanding of Media Manifest Core (MMC) www.movielabs.com/md/mmc and Media Entertainment Core (MEC) www.movielab.com/md/mec.

The approach defined in MMC is designed for implementations using both MEC and MMC are being used. This approach was chosen based in the input from multiple studios, platforms and service providers. Although designed to address the most common use cases, there are always exceptions. This document addresses those exceptions.

MMC describes the default method for delivering image. In MEC, Basic/LocalizedInfo/ArtReference contains an Image ID which maps to a corresponding Media Manifest Inventory/Image/@ImageID.

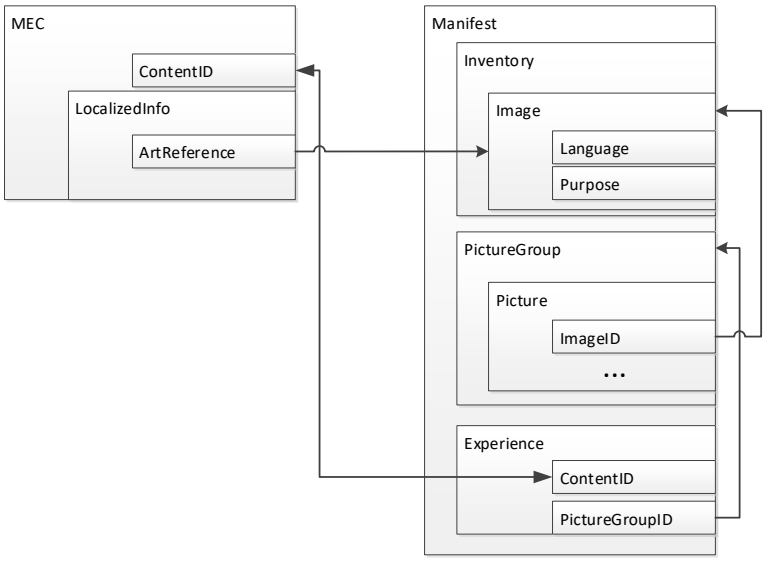
MMC also requires an instance of PictureGroups/PictureGroup that references the metadata images. That PictureGroup must be referenced from an Experience related to the title.

When MMC was developed, it was clear that MEC and MMC likely not be delivered together, and MMC could come before MEC or vice versa. Consequently, the linkage from MEC to MMC is just the reference in ArtReference.

Although it might not be apparent, MMC can stand alone. Start from Experience. Experience/ContentID is the metadata identifier. This indicates which MEC object applies. That MEC object then is very specific about which image apply where.

As mentioned above, that same Experience also references a PictureGroup which in turn references the metadata images. This instructs the platform to ingest the images, even if MEC has not yet arrived, associating those images with the ContentID.

There are various ways of encoding the PictureGroup, but for the purposes of this use case, they don't really matter. It ultimately maps to images which was the only requirement at the time. Below we'll provide some additional practices to provide information on the images.



2 When can Common Metadata or Media Manifest be used alone?

It is possible to deliver artwork files using either MEC (Common Metadata) or MMC (Media Manifest). As MMC is the preferred method for describing and delivering files, it is also the preferred method for delivering artwork files. However, Media Manifest was not designed for metadata so certain information can only be in MEC.

The simple answer is to let each tool do what it does best, but if that is not possible, there are alternatives. MEC can be used alone if there is a means to get from ArtReference to a file. As described in Section 5, ArtReference can reference online file locations (i.e., URLs) or file paths when the general location is known. If technical information or more information about file location is required, MMC should be used.

MMC can be used alone if the metadata the Inventory is sufficient (i.e., language and purpose). If other information, such as region is needed, MEC is necessary.

3 Manifest-only Image Delivery

It is possible to deliver images with only Media Manifest.

3.1 Encoding Inventory/Image

3.1.1 Language

In the Manifest, Inventory/Image contains some fields that describe the metadata image. The first is Image/Language. It says the language associated with the image. See the best practices on language on the Best Practices page www.movielabs.com/md/practices.

In most cases this is sufficient. However, be warned that it is not a complete solution because the language of the image is not necessarily a language of usage. MEC has LocalizedInfo/@language, LocalizedInfo/Region and other targeting values (e.g., TargetAudience). @language can reference images that have a different language. The simplest exception is textless images. These would be referenced in every @localizedInfo instance. Sometimes localizations share images because the precise language is not available. For example, an English image might be referenced by multiple localizations that have text but no images.

3.1.2 Purpose

Image/Purpose is a relatively new construct, but it's important when there are multiple images. Consider there are three sets of hero art. The images are distinguished by their purpose. In MEC it's ArtReference/@purpose. This corresponds with MMC's Image/Purpose.

3.2 Using Media Manifest to deliver only images

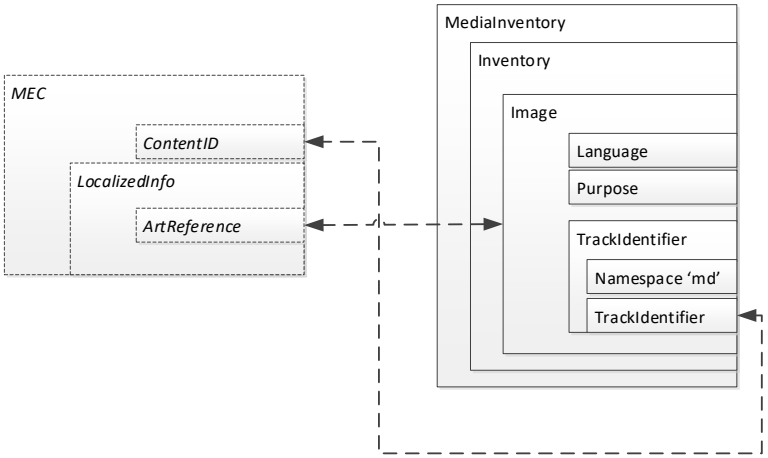
There are two ways to deliver only images with Media Manifest. The simplest method uses MediaInventory. A more complete solutions uses MediaManifest.

3.2.1 Using MediaInventory element

MediaInventory is a little-known element that exists in the Media Manifest schema. MediaInventory was created for delivering just the Inventory elements, with Images as a driving use case.

When delivering using MediaInventory, encode a MediaInventory/Image instance for each image. As Image contains information about language and purpose, this might be enough. However, it has no specific reference to the content for which it applies.

The following illustration shows the essential elements of delivering images with MediaManifest.



You can use the following additional values to associate the image with content

- MediaInventory/Compatibility/SpecVersion = version of Media Manifest specification. This typically corresponds with the version in the namespace.
- MediaInventory/Compatibility/Profile = "MI-IMAGE-1"
- MediaInventory/Image for each image
    - Image/Purpose = purpose of image, if applicable
    - Image/Language = language of image. One instance for each language in the image. Note that this is a description of the image, not instructions on where to use image.
    - Image/TrackIdentifier is included to reference ContentID (or some other bilaterally agreed-upon identifier). For ContentID,
        - TrackIdentifier/Namespace = 'md'
        - TrackIdentifier/Identifier = content ID associated with image
    - Image/Width, Height, Encoding are required fields
    - Image/AssetIntent, if region is being specified
        - Type = "Region" or Type= "ExcludedRegion"

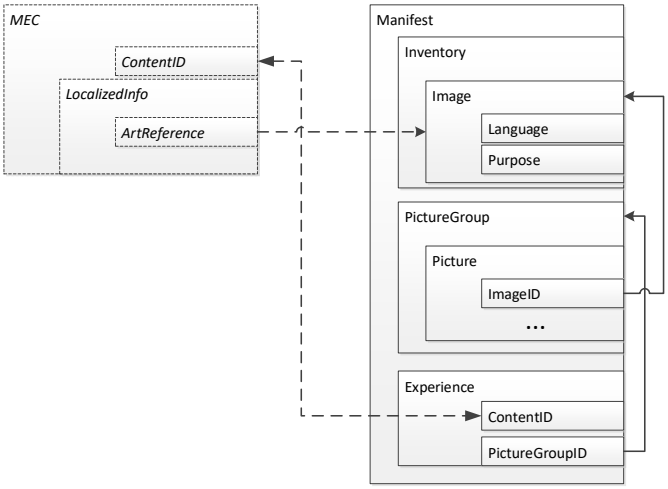
- SubType = value of country or country Region from Common Metadata, Section 3.2. For example, “us”. One instance for every Region or Excluded Region.

- All other fields can be encoded as appropriate

3.2.2 Using MediaManifest element

The basic method for delivering images in Media Manifest is described in MMC. However, this is in the context of delivering everything. This describes how to deliver *only* images.

The following illustration shows the essential elements of delivering images with MediaManifest.



Encode as follows

- MediaManifest/Compatibility/SpecVersion = version of Media Manifest specification. This typically corresponds with the version in the namespace.
- MediaManifest/Compatibility/Profile = “MM-IMAGE-1”
- Inventory/Image encoded as described for MediaInventory
- PictureGroups/PictureGroup included to contain images (same as MMC)
  - @PictureGroupID can be anything, preferably conforming to md: naming conventions
  - One instance of Picture
    - Picture/PictureID can be anything, preferably conforming to md: naming conventions.
    - Picture/ImageID – on instance per image. Each instance contains ImageID of image
- Include an Experience

- Experience/ContentID is ContentID associated with metadata. If this is not known, then use some ID that will be mutually understood to associate the image with the metadata
- Experience/PictureGroupID is the Picture Group ID for the PictureGroup

Following is an example of a Picture element that reference 5 hero images. There are three variants of "hero2". (Note that not a full encoding.)

```
<Picture>
    <PictureID> md:pictureid:eidr-x:abcd-abcd-abcd-abcd-e:hero</PictureID>
    <ImageID>md:pictureid:eidr-x:abcd-abcd-abcd-abcd-e:hero1</ImageID>
    <ImageID>md:pictureid:eidr-x:abcd-abcd-abcd-abcd-e:hero2</ImageID>
    <ImageID>md:pictureid:eidr-x:abcd-abcd-abcd-abcd-e:hero2-alt1</ImageID>
    <ImageID>md:pictureid:eidr-x:abcd-abcd-abcd-abcd-e:hero2-alt2</ImageID>
    <ImageID>md:pictureid:eidr-x:abcd-abcd-abcd-abcd-e:hero3</ImageID>
</Picture>
```

The corresponding Inventory record for hero2, alternate 1 might look like the following. All the hero2 variants will have the same metadata indicating they are equivalent.

```
<Inventory>
    <Image ImageID ="md:pictureid:eidr-x:abcd-abcd-abcd-abcd-e:hero2-alt1">
        <Purpose>hero2</Purpose>
        <Width>600</Width>
        <Height>800</Height>
        <Encoding>image/jpeg</Encoding>
        <Language>en</Language>
    <Image>
</Inventory>
```

4 Submitting multiple image variants

This practice addresses the use case of delivering multiple equivalent images and letting the recipient choose which images they want.

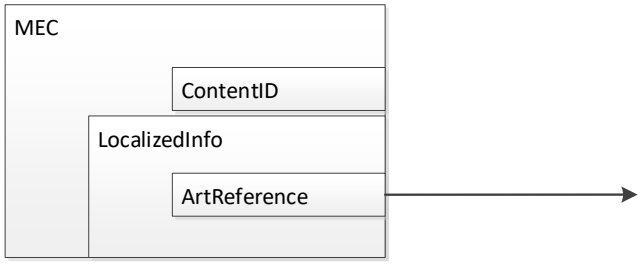
Whether using MEC or MMC, there is no prohibition from delivering multiple equivalent images. If multiple images are "equivalent" they should have the same (MEC) LocalizedInfo/ArtReference/@purpose or (MMC) Inventory/Image/Purpose.

When we say images are equivalent, we are saying they can be substituted. Most typically, they are variants of images with the same placement. For example, an interface might have a space for an 3:4 image with cover art. A studio might supply three images, any of which can go in that space; such as, different images with different actors. These images would be considered equivalent.

5 Delivering Images with only MEC

We previously describe methods for delivering images using MMC. Of course, images need not be delivered in this manner. If there is a way to associate images with the metadata (eventually), it should work. Connecting MEC to images is done by aligning ArtReference with your delivery method.

The following illustrated MEC-only image delivery:



MMC contains mechanisms for referencing file location, but there is no equivalence in MEC. Consequently, ArtReference must be encoded with information sufficient to resolve the images file's location. For example,

- ArtReference is a URL where images can be retrieved directly.
- ArtReference is a filename corresponding with the delivery methods (e.g., /resource/greatmovie/images/image235.png)

5.1 Delivering Common Metadata with only images

A special case of delivering images with only MEC is delivering only image (i.e., no other metadata). This practice is designed to address the use case where there is no localized metadata, only images.

The practice is very simple. Exclude everything that's not LocalizedInfo/ArtReference. There are a couple of required fields. These can be included or not.

If using a version of Common Metadata that has Basic/Compatibility,

- Compatibility/Profile= "CM-IMAGE-1"

The Media Entertainment Core specification includes elements for delivering only LocalizedInfo called LocalizedInfoDelivery and LocalizedInfoDeliveryList. These can be used in lieu of providing a CoreMetadata element. The former is used when only one LocalizedInfo is to be delivered. The latter for multiple.

6 Sample scenarios

Following are a few scenarios that illustrate image delivery

6.1 Scenario 1 – Separate delivery of metadata and artwork

This scenario has separate metadata and artwork deliveries. Images are coming from a separate source than images.

- Delivery 1: Metadata delivered
    - Metadata is delivered in MEC as usual, but there would be no ArtReference instances
- Delivery 2: Artwork delivered
    - Artwork is delivered using Manifest as described in 3.2.
    - It is essential that ContentID matches between MEC and Manifest to the artwork can be associated with the corresponding metadata.
    - It is also important that the Manifest's Image/Language match MEC's LocalizedInfo/@language to associate the image to the correct language metadata.
    - Note that images in Manifest cannot be associated to metadata for a particular region. There is no Manifest equivalent to LocalizeInfo/Region.

6.2 Scenario 2 – Late delivery of artwork details

Scenario 1 comes in three deliveries:

- Deliver 1: 1000 titles, each with their own Content ID.
    - MMC is delivered, but without images.
    - MEC is delivered, but without ArtReference populated.
- Delivery 2: 3,000 images delivered, tagged with Content IDs.
    - MMC is delivered. See Section 3.2 for specifics.
- Delivery 3: Information is provided regarding the language and territory of each image.
    - MEC is delivered with only image data (Section 5.1). LocalizedInfo instances include @language and Region (as appropriate), and ArtReference references the ImageID from the previous step.

A variant on Scenario 2 is that Delivery 2 includes MEC, but with incorrect information. Delivery 3 corrects the misinformation in Delivery 2. This is not ideal but could be processed correctly as long as @updateNum in MEC is correctly encoded to indicate the update.

6.3 Scenario 3 – Updating images

"I want to be able to update either metadata or artwork for a title without resubmitting the whole title package."

Metadata is updated by sending a MEC update. Note that it is important that @updateNum is incremented to allow the recipient to determine which MEC instance is the most recent.

Similarly, images can be updated by sending Manifest updates. This can be done by sending the complete Manifest or by using the MediaManifestEdit element. With MediaManifestEdit, old images can be removed by including an instance of DeleteObjects/ImageID, each with an ID for the image to be deleted. Images can be added by including a AddObject/Inventory/Image instance for each image to be added.

7 Media Manifest-only Image Delivery, non-metadata images

Other images can also be delivered using these methods. For example, in the platform supports it, cards (e.g., dub cards, rating cards) can be delivered as images. This would follow the same practice defined in MMC, except that instead of using the Clip element in PlayableSequence, use ImageClip with a reference to the ImageID of the card image.

The practice is to encode MediaInventory with one or more properly encoded Image elements.

Note that this does not preclude the use of other elements being included.