

# Best Practice: Localization Updates

This document provides best practices (recipes) for sending metadata and track localizations. This is especially important in workflows where localizations come from different sources than the original content.

Applicable to these practices are: Avails, Common Metadata, Media Entertainment Core (MEC), Media Manifest, Media Manifest Core (MMC), Cross-Platform Extras (CPE) and other MovieLabs Digital Distribution Framework (MDDF) Specs. This document assumes familiarity with the referenced specifications.

Version	Date	Description
1.0	July 26, 2017	Initial publication. Derived from Best Practices for Delivery



This work is licensed under a [Creative Commons Attribution 3.0 Unported License](https://creativecommons.org/licenses/by/3.0/).

**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.

## References

[CM]	Common Metadata, TR-META-CM, <a href="http://www.movielabs.com/md/md">www.movielabs.com/md/md</a>
[Manifest]	MovieLabs Common Media Manifest Metadata v1.4, TR-META-MMM, <a href="http://www.movielabs.com/md/manifest">www.movielabs.com/md/manifest</a>
[Avail]	EMA Content Availability Data (Avails), TR-META-AVAIL, <a href="http://www.movielabs.com/md/avails">www.movielabs.com/md/avails</a>
[MEC]	Media Entertainment Core, TR-META-MEC, <a href="http://www.movielabs.com/md/mec">www.movielabs.com/md/mec</a>

---

## 1 LOCALIZATION UPDATE

The most frequent update is the addition of a language. This section describes the rules for performing a localization update. Our intent is to provide step-by-step recipes for encoding.

First we describe how to update metadata by itself. This is useful for opening storefronts or providing library views in new territories or simply to reach a broader audience within a territory. Then we describe how to update both metadata and track data. This supports fulfillment where additional tracks are required. The same process can apply whenever any new track is available.

It is important that the entity sending the update can construct identifiers for objects being updated. This requires either prior knowledge of the identifiers, such as having a current version of the manifest, or having the ability to construct identifiers. The ID naming conventions in this document can help in many cases, but may require additional naming convention agreements between parties.

Media Manifest and, soon, other specs support @updateDeliveryType to provide the recipient with an indication of the specifics of a delivery. In this case, if the @updateDeliveryType is 'Localization 1a', it means only metadata is provided. If it's 'Localization 1', it means both metadata and tracks are being provided.

While updates can be performed using the Media Manifest, the most straightforward method is to use MedianInventory and MediaManifestEdit.

In updates and issue resolution, it is possible that only the Inventory changes. To facilitate delivering the Inventory alone, there is a MediaInventory element. The essential criteria are that all references to the Inventory are the same.

For example, consider an update where a defective track is replaced. The only change is some combination of TrackIdentifier and ContainerReference. The track reference (TrackID) from a Presentation is not affected.

### 1.1 Localization Update – Metadata only (localization a1)

Localization 1a is designed to add only metadata to an existing Experience, such as adding region-specific artwork. This will typically result in an updated Experience that includes the new regions. If metadata is included in the Experience, it is necessary to update the Experience and add Inventory for the images. If the Experience references metadata through ContentID (i.e., metadata delivered separately), it is only necessary to include the image. A Localization 1a update can be as simple as an Inventory with an image.

Localization 1a has the same rules as Localization 1, with the following exceptions in MediaManifestEdit:

- @updateDeliveryType='Localization1a'
- MediaManifestEdit/Inventory is present if necessary.

- Image is included as appropriate
- MediaManifestEdit/Presentations/Presentation is not present
- MediaManifestEdit/PlayableSequences is not present

## 1.2 Localization of tracks and metadata (localization 1)

Localization 1 and its variants involved sending *MediaManifestEdit*. Take care not to use *MediaManifest* instead.

The following rules instruct how to populate *MediaManifestEdit* to update Media Manifest using the Localization 1 model:

- *MediaManifestEdit@updateNum* must reflect an update. That is, it must be greater than all previous *@updateNum* values. If by prior agreement, the sequence of updates is understood, this can be waived.
- *MediaManifestEdit@updateDeliveryType*=‘Localization1’
- Any data referenced by *MediaManifestEdit/DeleteObject* will be removed prior to adding objects
- *MediaManifestEdit/AddObject/Inventory* is present.
  - For each language added
    - At least one Audio or Subtitle element must be present. In this model, there is no such thing as a language update without audio or subtitle.
    - All audio and subtitle tracks for the language that are included in a Presentation must be included in the Inventory
    - All track identifiers must be unique. That is, they must be distinct from each other and must not exist in any previous version of the manifest.
  - Other Audio, Video and Subtitle tracks are included as appropriate. For example, localized ratings or anti-piracy pre-roll video might be required as part of a localization.
  - Image and Interactive are included as appropriate
- *MediaManifestEdit/AddObject/Presentations/Presentation* must be present
  - A Presentation must be included for all Presentations that include new tracks.
  - To add tracks to an existing Presentation
    - *Presentation/@PresentationID* must match the Presentation to be updated

- 
- TrackMetadata/TrackSelectionNumber must match existing Track Metadata
  - VideoTrackReference, AudioTrackReference and SubtitleTrackReference are only included for new tracks.
  - LanguagePair may be added
  - Chapters may not be modified in any way
  - If an object (e.g., a track reference in a Presentation) in the update already exists in the original object, it is ignored.
  - Note: There is currently no way to delete a track
  - MediaManifestEdit/AddObject/PlayableSequences/PlayableSequence may be added.
  - MediaManifestEdit/AddObject/Experiences/Experience can be added.
    - It must have a unique ExperienceID
    - @updateNum='1'
  - MediaManifestEdit/Experiences/Experience can be modified as follows:
    - @ExperienceID must match the Experience being updated
    - @updateNum must reflect an update
    - Language and/or Region must be included, these are assumed to be in addition to existing Languages or Regions covered. The following defines how languages are regions are added. The rules are the same for Language and Region.
      - Case 1: All languages/regions are already covered. This is represented by the absence of Language/Region and ExcludedLanguage/ExcludedRegion. Adding a language/region has no effect because they are already implicitly included.
      - Case 2: Language/Region instances exist for other languages/regions. An instance of Language/Region is created for the language/region.
      - Case 3: ExcludedLanguage/ExcludedRegion instance exists for language/region. Adding a language/region requires the instance be removed. By definition if it's not Excluded, it's included.
    - If there are new Playable Sequences or Presentations, Audiovisual must be modified or created as follows
      - If @ContentID matches @ContentID in an earlier version of the Experience, this Audiovisual replaces the earlier version

- If @ContentID does not match @ContentID in an earlier version, Audiovisual is added.
- MediaManifestEdit/AddObjects/BasicMetadata must be included for localizations not already in metadata. Note that if a region is added with a language that is already supported, BasicMetadata might not be needed.
  - @ContentID is required
  - WorkType is required
  - UpdateNum should be included and reflect and update. However, it is expected that updates are provided by uncoordinated parties so it may not be practical to enforce proper UpdateNum usage.
  - LocalizedInfo must exist for each language
    - @default should not be included. Generally, there should already be an instance of LocalizedInfo with @default='true'. If the default LocalizedInfo must be changed, this can be handled through an Instructions-only Special Delivery Condition (**Error! Reference source not found.**).
  - RatingSet/Rating should be included for covered regions, as applicable
  - People should be updated if localized information is included, such as Job/JobDisplay and Name/DisplayName
  - Other elements and attributes need not be included.
- If an update cannot be accomplished within these rules, a complete Media Manifest must be provided.

As an alternative, Media Entertainment Core (MEC) can be used. MEC does not have a means to signal that this delivery is an update so the parties must determine this from context. The following rules apply to updating metadata via MEC in Localization 1:

- BasicMetadata should include all new localizations. Rules are the same as for MediaManifestEdit/AddObject/BasicMetadata above.
- If DigitalAsset is included in metadata delivery it must include new track definitions. Note that DigitalAsset is redundant with Inventory and will likely not be included.
- TitleInternalAlias should be included
- TrackingID must be included as appropriate
- Source should be included
- CompanyDisplayCredit and GroupingEntity should be included if localized data differs from what was previously delivered (of if there is any doubt).

DRAFT