

Playlists

Encoding Best Practice

This practice defines the method for describing a Playlist in Media Manifest. Playlists are sequences of distinct works that play back-to-back, such as a promo followed by a recap followed by an episode.



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.

REVISION HISTORY

Version	Date	Description
Draft	February 17, 2023	Review draft
1.0	December 4, 2025	Initial publication

1 INTRODUCTION

1.1 Scope

This document describes a method to describe sequences of loosely related audiovisual material that play back-to-back. This might include a promotional video followed by an episode recap followed by the episode itself. For the context of this paper, we call this a Playlist.

Tightly coupled audiovisual material, such as cards, are Playable Sequences, not Playlists and are implemented through a Manifest's Playable Sequence mechanism.

1.2 Context

There are three levels associated with playing one bit of audiovisual material after another. Two are documented the Playable Sequence section of Media Manifest (Section 5.2). The third is defined here.

Playable Sequence mechanisms are intended for pre-roll and post-roll that are semi-permanently attached to the main content. This primarily includes cards (i.e., rating, anti-piracy, health, dub, etc.), but can also include other elements, such as distribution logos.

Playable Sequence makes a distinction between clips that are seamlessly stitched (i.e., the content plays the next clip immediately after the other), and content that is separate by some period of black and silence. These are the first two. Note that IMF CPLs also play seamlessly, and this is the preferred method for seamless stitching.

The third method is Playlists. At some point in the distribution chain someone decides that content should be played one after another. For example, one might play a recap of previous episodes before showing a new episode. Another example is a promotional video that plays before or after the feature.

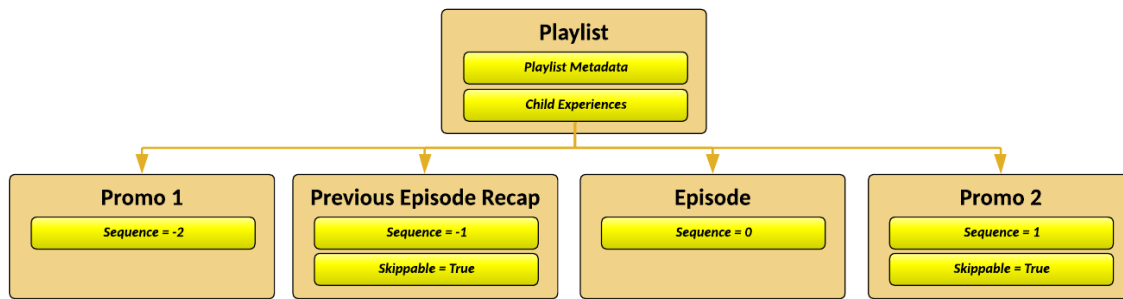
As an aside, there is an important fourth, dynamic ad insertion, and possibly others; but that's beyond the scope of this paper.

2 PLAYLISTS

2.1 General Approach

The recommended method for Playlists is to create an Experience for each element of the Playlist. Then create one or more Experience instances that reference those Experiences.

In this paper, we will reference the following structure. These are Experiences organized as a Playlist.



2.2 Playlist Experience Structure

A Playlist consists of a parent Experience which defines the structure of the playlist and two or more child Experiences and reference the content of the Playlist.

Child Experiences each are referenced by the ChildExperience element in the Playlist Experience.

Within the Playlist Experience, Experience/Type = “Playlist”.

In the example above, the Playlist consists for a promotion (Promo 1), a recap, an episode, and another promo.

2.3 Order of Playback

Each child experience will be played in the order defined by Experience/ChildExperience/Sequence/Number.

Playlists should play in order regardless of what the sequence numbers are, or even if there are gaps. However, it is strongly recommended that sequence numbers are contiguous, and the feature has a value of zero (0).

2.4 Multiple Manifests

It is generally preferable to include separate Manifests for each element in the Playlist. For example, the promo would have a Manifest, the recap would have a Manifest, and so forth.

If external Manifests are used, ChildExperience/ExternalManifest must be used.

2.5 Localization

The same rules apply to the Playlists Experience as apply to other Experiences. If Experience/Language|ExcludedLanguage or Experience/Region|ExcludedRegion are set, that guidance should be followed. This is documented in Manifest, section 8.2.

One option is to specify these values for the Playlist Experience. Then that Experience will only apply to the region(s) and language(s) specified.

Another option is to specify language and/or region in the child Experiences. If this is done, it is necessary to provide adequate coverage by including multiple Experiences. Where there are multiple child Experiences to choose from, they should all have the same Sequence/Number.

2.6 Skippable Experiences

A skippable Experience is one that the user may optionally skip, typically from a button that says “Skip”.

Experiences that cannot be skipped will play without an option to skip. Note that this does not imply that other mechanisms, such as fast forward, cannot be used. That is up to the player policy.

Skippable Experiences are signaled by including an instance of Experience/Terms with `@termName = “skippable”`.

If skipping is allowed after a specific delay, include an Experience/Terms instance with `@termName = “skipwait”` and `Decimal = delay period in seconds`. Zero means segment is skippable immediately.

Note that earlier versions of this document recommended setting an instance of Experience/SubType to “Skippable”. This should be used on older versions of Manifest that do not support Experience/Terms.

2.7 Windows

The Playlist does not directly support start (e.g., embargo) or end times for Playlists.

If this feature is required, please communicate this to the authors.

2.8 Dynamic Range

The Playlist may include a playback experience in multiple dynamic ranges. As an example, a Playlist may include both High Dynamic Range (HDR) and Standard Dynamic Range (SDR) video. The recommended approach for Playlists supporting multiple dynamic range video formats is to include video tracks for each dynamic range in a single Presentation.

As is the practice for audio, the appropriate video track must be selected based on entitlement, security, device capabilities, and other platform-specific requirements. Selection of the appropriate track may use information in Inventory and, if present, `@priority` in track references in the Presentation.

2.9 Markers in Playlist

Each element in a Playlist can optionally include Markers in the Media Manifest’s Presentation. Once all elements are playlisted, the full timeline of markers is the responsibility of the consuming application.

Markers are tied to the timeline of the Presentation. When Presentations are concatenated by the Playlist mechanism, Marker times may need to be mapped to the Playlist timeline. If the consuming application needs to create a Playlist timeline it should take the timecode of each Presentation's Markers and construct a new timeline with the Markers properly aligned and offset based on the sequence of playback. Note that given the complexities of timecodes, this is not necessarily a simple calculation.

3 AVAILING PLAYLISTS

Playlists are availed through ALIDExperienceMap where the ALID is in the Avail, and the Experience is the Playlist Experience.

The Playlist Avail supersedes any Avails that may exist for child Experiences. Be careful.

It is recommended that the Avail superseding another Avail reference the ALID for that Avail. In XML this is done in the AssociatedALID element in with “isplaylistroot” in @relationship. In Excel, use RefALID (relationship is implicit).