

Cards Encoding Best Practice

This practice defines how to encode cards, such as anti-piracy cards, rating cards, health cards and dub cards.

EARLY DRAFT PUBLISHED FOR COMMENTS



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
1.0	TBD	Initial publication

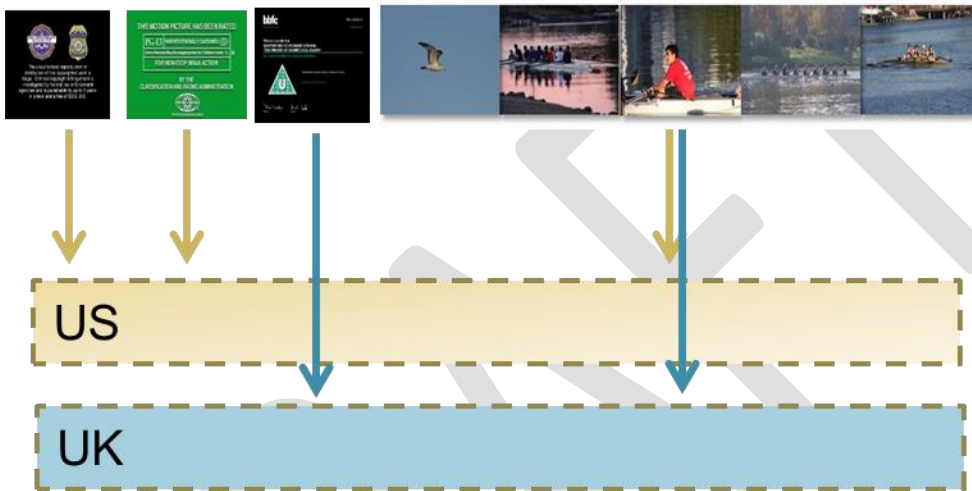
1 CARDS

A Card is a short segment of video that either precedes or follows a feature in playback.

The following images represent antipiracy and rating cards:



The goal is to construct the appropriate playback from cards and main (feature) video:



1.1 Motivation

There are three primary motivations for the way we handle cards:

- In each context (e.g., territory), ensure the correct cards are played at the appropriate time.
- Avoid playing cards that need not be played. Examples of this done badly is stitching all dub cards and/or antipiracy cards to the end of a video.
- Reduce the total size of video encodes. More specifically, avoid an end-to-end encode of video unique to each territory.

Ultimately, we expect playback to dynamically select the appropriate card at playback time. This allows a single main video to cover multiple territories and languages. We provide the mechanisms to allow players to select the correct territory-specific cards to be played—typically as pre-roll—and language-specific cards—typically played as post-roll.

Note that it is ultimately up to the platform to decide whether cards are bound to main video at encoding time or selected at playback time.

1.2 References

Cards are described and defined in the following specifications. This best practice assumes the reader's familiarity with these documents with particular focus on the sections listed:

- Common Metadata, primarily section 5.2.1.1
- Media Manifest, primarily section 5.2.1.1
- Media Manifest Core, Section 2.1.1, 3.2.4

2 CARD ENCODING

2.1 Card encoding in Inventory

Inventory/Video and Inventory/Subtitle contain CardSetList objects. These objects are designed to capture which cards are in the video or timed text.

For example, consider the following videos; one with cards and one without



The former has no CardSetList. The latter has a CardSetList listing the cards.

From the Inventory data indicates that the first might need antipiracy and rating cards and the second already has these cards. Also, it indicates the second has US cards and would likely be unsuitable for other territories.

2.2 Card encoding in Presentation

When we talk about cards, we are assigning cards to a specific segment of the playback timeline. If a card is embedded in a video, then the rest of the timeline must contain track essence that corresponds with that portion of the timeline (even if the essence is silence or absence of text). That is, if there is card video, other tracks in the Presentation must be conformed to the same timeline. In other words, Cards don't just exist in one track, they also exist across a Presentation.

The Presentation has the means to signal some card information through Markers (Media Manifest, Section 5.1.5). Markers indicate the start and end of events. Here are some examples:

- “FFCL” is First frame of Company/Production Logo.
- “FFDL” is First Frame of Distribution Logo.
- “FFOB” is First Frame of Rating Band.
- “FFOC” is First Frame of Composition”
- “FFDC” First Frame of Dubbing Credits.

These can be extended as needed.

2.3 Card encoding in Playable Sequences

This should be pretty much covered in Media Manifest and Media Manifest Core (MMC). Please the sections of these documents noted in References above.

The main feature is given sequence = 0. Anything played before the main feature is given sequence < 0, and anything after is given sequence > 0.

If there is a choice of cards based on territory or language, they are given the same sequence. For example, typically all dubbing cards have a sequence of 1. They immediately following the main video.

Choosing between cards can be based on language, territory or both.

For language, the player will select the PlayableSequence/Clip whose @language matches the playback language. For example, if the user is playing es-419 audio and there is a dub card whose PlayableSequence/Clip/@language='es-419', this Clip is played after the feature.

For territory, Clip/Region encodes territories where the Clip applies. For example, if it is a US Homeland Security card, then Clip/Region='us'. If playback is occurring in the US, then this Card is selected.

There should not be more than one card that matches language and territory for the same sequence. The player would not be able to determine the correct card.

3 CARD USE CASES

This section provides specific instructions around given Use Cases

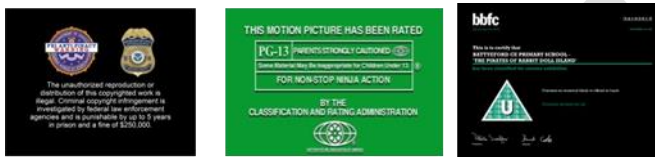
3.1 Cards delivered as separate assets

This use case covers cards delivered as their own audiovisual assets.

This will typically include one main video:



And a collection of cards:



In this situation, each Clip corresponds with a single Presentation.

3.2 Cards delivered embedded in video

This use case covers cards stitched to feature video.

Consider the following example. Cards are stitched to the main feature.



This example has US Cards and is playable in the US. In this case, there need not be a Playable Sequence. However, the Experience that references this Presentation must be localized to the US (i.e., Experience/Region='US').

3.2.1 Extracting Main Presentation and Cards

Although a Presentation that includes Cards is playable in a given territory, it will not meet the efficiency goals articulated in the opening section. In this model, one must maintain a distinct full-length asset for each combination of cards. If dub cards are stitched top the end, the user will be subjected to the full collection of dub cards, rather than just the card that applies to their language.

It is therefore desirable to have the ability to extract cards from such as Presentation.

3.2.1.1 Inventory

There are Inventory entries for all tracks in the composition.

CardSetList must be correctly encoded in the Inventory/Video instance referenced in the Presentation.

In the example above, there is a single CardsetList instance. The first has Region of “US”, and two Cardset instances (one for each Card). The first Cardset has Type = “AntiPiracy” and Sequence = “1”. The second Cardset has Type = “Rating” and Sequence = “2”.

[CHS: better to include XML example]

3.2.1.2 Presentation

There is a Presentation for the composition. It references the tracks in the Inventory. There is nothing specific to Cards here, although Markers may be set if desired.

3.2.1.3 Playable Sequence

There is a Playable Sequence for the composition. It must indicate where the Card is found in the composition, and describe the Card.

Each Card is defined in a PlayableSequence/Clip as follows

- EntryPointTimecode and ExitPointTimecode correspond to the start and end of the Card
- Purpose/WorkType is the Card type as defined in Common Metadata Section 5.2.11.2. In the example, above, “AntiPiracy” would be used for the first Clip and “Rating” would be used for the second.
- CardsetSeq is set to the value that corresponds with the CardsetList/Card instance in the referenced Inventory/Video (via the Presentation). As stated above in the example, the anti piracy card has a Sequence of “1” and the rating card has a Sequence of “2”.

Assuming the Playable Sequence represents the required playback sequence, the next Clip could be for the main composition.

[CHS: Include XML example]

3.2.1.4 Generalizing

The mechanism described above is more general than it sounds (and the example implies). Specifically, the previous example assumes that the playback sequence is the same as the encoding sequence and the only purpose of this process is the extract the cards. However, there are more use cases.

Consider if the cards are delivered as a single conformed composition consisting of these cards.



The Inventory/Video/CardsetList describes each Card. There is a Presentation for this composition, and another Presentation instance for the feature (which might include other cards).

And, there is a Playable Sequence for the final composition. This Playable Sequence would also include three Clips (one for each card, and one for the main composition). There are two main differences

- The Card Clips reference the Card Presentation (different from the feature Presentation)
- The Video/CardsetList/Cardset/Sequence values and the Clip/@sequence are no longer ordered in the same way. In the original, Clip @sequence = -2 corresponded with Card Sequence = 1, and @sequence = -1 corresponded with Sequence = 2. In the new example, Clip @sequence = -2 corresponds with Card Sequence = 2, and @sequence = -1 corresponds with Sequence = 1. In a general sense, it could come from a complete different composition.

The point is that the only way to match a Clip to a Card is to match the Clip's CardsetSeq to the Card's Sequence.

3.2.1.5 Specialized Simplifying Use Case

This section describes a simplification that is possible where one can always assume:

- Clips with @sequence not equal to 0 are cards
- The card are sequenced in the composition in the same order as the Playable Sequence.

In this case, CardsetSeq for the first reference to a Clip within a Presentation defaults to 1, and subsequent references increase by one.

Returning the original use case,

- Clip -2 corresponds with Card Sequence 1
- Clip -1 corresponds with Card Sequence 2

If there were dub cards

- Clip 1 corresponds with Cards Sequence 3
- Clip 2 corresponds with Card Sequence 4
- etc.

If you are feeling particularly brave, you can reference more than one composition (i.e., more than one Presentation). Assuming dub cards are in a different composition:

- Clip -2 corresponds with Presentation 1, Card Sequence 1
- Clip -1 corresponds with Presentation 1, Card Sequence 2
- Clip 1 corresponds with Presentation 2, Cards Sequence 1
- Clip 2 corresponds with Presentation 2, Card Sequence 2
- etc.