**Asset Ordering, Delivery and Tracking**

**Contents**

1 Introduction 1

1.1 Overview 1

1.2 Document Organization 3

1.3 Document Notation and Conventions 3

1.3.1 XML Conventions 4

1.3.2 General Notes 5

1.4 Normative References 5

1.5 Informative References 6

1.6 Best Practices for Maximum Compatibility 6

2 Structure 7

2.1 Requirements Structure 7

2.1.1 Scope 7

2.1.2 Profiles 7

2.1.3 Profile Examples 9

3 General Types Encoding 12

3.1 Simple Types 12

3.1.1 RangeCondition 12

3.2 DeliverySource-type 12

3.3 DeliveryHandling-type 13

3.4 DeliveryInstructions-type 13

3.5 Terms 13

3.6 DeliveryParams-type 15

3.6.1 DeliveryResolution-type 15

3.7 Types that reference objects directly 16

3.7.1 DeliveryObjectReference-type 16

3.7.2 DeliveryFileReference-type 17

3.8 Types that reference objects through description 18

3.8.1 DeliverObjectDescription-type 18

4 Profiles 19

4.1 Administrative Profile 19

4.2 Language Profiles 19

4.3 Artwork Profiles 20

4.3.1 ArtworkProfile-type 20

4.3.2 ArtworkImage-type 21

4.4 Technical Profiles 21

4.4.1 TechnicalProfile-type 22

4.4.2 Interpretation of terms within Technical Profiles 23

4.4.3 TechAudio-type 24

4.4.4 TechVideo-type 25

4.4.5 TechSubtitle-type 28

4.4.6 TechCard-type 29

4.4.7 DeliveryImage-type 29

4.4.8 TechMetadata-type 30

4.4.9 TechContainer-type 31

4.5 Product Profiles 32

4.5.1 ProductProfile-type 32

4.5.2 ProductProfileInfo-type 32

4.5.3 DeliveryPromotional-type 33

4.5.4 DeliverySupplemental-type 33

5 Content Delivery Requirements 35

5.1 GeneralRules-type 35

5.2 TerritoryRules-type 36

5.2.1 CategoryRules-type 37

6 Asset Order 37

6.1 DeliveryRequest-type 37

7 Asset Status Manifest (ASM) 38

8 QC Report 38

8.1 QCReport-type 38

8.2 Error types 39

8.2.1 QCError-type 39

8.2.2 QCErrorDescription-type 39

8.2.3 QCCategoryError-type 40

9 Asset Status Manifest (ASM) 43

10 Asset Status 43

11 Avail Confirmation 44

12 NOTES 45

[Creative Commons License](http://creativecommons.org/licenses/by/3.0/)  
This work is licensed under a [Creative Commons Attribution 3.0 Unported License](http://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 Common Metadata. Whether to adopt the Common Metadata 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 the Common Metadata for any purpose, and any liability for any damages or other harm you may incur as a result of subscribing to this Common Metadata. **Revision History**

|  |  |  |
| --- | --- | --- |
| **Version** | **Date** | **Description** |
| 1.0 |  | Original Version |

# Introduction

This document defined data used in the delivery of assets, within the MovieLabs Digital Distribution Framework (MDDF). The following illustration shows the MDDF flow, with Asset Ordering and Delivery data shown in purple.



This specification is designed to work with other MDDF specifications or with proprietary/legacy specifications.

## Overview

The Asset Ordering and Delivery Process is addressed in three parts

* Rights Management – Generation and delivery of Avails or Title List
* Asset Planning – All processes associated with determining which assets (audio, video, subtitles, artwork, metadata, etc.) will be delivered
* Asset Delivery – Processes associated with the delivery of assets

These are illustrated in Figure 1 below.

The Rights Management process is covered by Avails and is not further discussed in this document. See [www.movielabs.com/md/avails](http://www.movielabs.com/md/avails) for more information.

Asset Planning is further divided into asset policies that span Avails, and Avail-specific or titles-specific asset selection. Asset policies are captured in “Content Delivery Requirements”. Avail or title-specific requests are included in Avail Confirmations, Asset Orders, and Asset Status Manifests.

Asset Delivery has several parts including a Media Manifest Core (MMC) delivery spec, the assets themselves, QC failure reports that document issues, and Asset Status information. MMC is documented elsewhere ([www.movielabs.com/md/mmc](http://www.movielabs.com/md/mmc)), and this specification is neutral to assets delivered—we attempt to support almost any format. This specification documents the QC failure reports and Asset Status data.

Figure 1: Asset Distribution Workflow



## Document Organization

This document is organized as follows:

1. Introduction—Provides background, scope and conventions
2. Asset Planning and Delivery
3. General Types Encoding
4. Content Delivery Requirements
5. Asset Order
6. Asset Status Manifest (ASM)
7. QC Fail Report
8. Asset Status
9. Avail Confirmation [CHS: this belongs in Avail]

## Document Notation and Conventions

As a general guideline, the key words “MUST”, “MUST NOT”, “REQUIRED”, “SHALL”, “SHALL NOT”, “SHOULD”, “SHOULD NOT”, “RECOMMENDED”, “MAY”, and “OPTIONAL” in this document are to be interpreted as described in [RFC2119]. That is:

* “MUST”, “REQUIRED” or “SHALL”, mean that the definition is an absolute requirement of the specification.
* “MUST NOT” or “SHALL NOT” means that the definition is an absolute prohibition of the specification.
* “SHOULD” or “RECOMMENDED” mean that there may be valid reasons to ignore a particular item, but the full implications must be understood and carefully weighed before choosing a different course.
* “SHOULD NOT” or “NOT RECOMMENDED” mean that there may be valid reasons when the particular behavior is acceptable, but the full implications should be understood and the case carefully weighed before implementing any behavior described with this label.
* “MAY” or “OPTIONAL” mean the item is truly optional, however a preferred implementation may be specified for OPTIONAL features to improve interoperability.

Terms defined to have a specific meaning within this specification will be capitalized, e.g. “Track”, and should be interpreted with their general meaning if not capitalized.

Normative key words are written in all caps, e.g. “SHALL”.

Normative requirements need not use the formal language above.

### XML Conventions

XML is used extensively in this document to describe data. It does not necessarily imply that actual data exchanged will be in XML. For example, JSON may be used equivalently.

This document uses tables to define XML structure. These tables may combine multiple elements and attributes in a single table. Although this does not align with schema structure, it is much more readable and hence easier to review and to implement.

Although the tables are less exact than XSD, the tables should not conflict with the schema. Such contradictions should be noted as errors and corrected.

#### Naming Conventions

This section describes naming conventions for Common Metadata XML attributes, element and other named entities. The conventions are as follows:

* Names use initial caps, as in InitialCaps.
* Elements begin with a capital letter, as in InitialCapitalElement.
* Attributes begin with a lowercase letter, as in initiaLowercaseAttribute.
* XML structures are formatted as Courier New, such as md:id-type
* Names of both simple and complex types are followed with “-type”

#### Structure of Element Table

Each section begins with an information introduction. For example, “The Bin Element describes the unique case information assigned to the notice.”

This is followed by a table with the following structure.

The headings are

* Element—the name of the element.
* Attribute—the name of the attribute
* Definition—a descriptive definition. The definition may define conditions of usage or other constraints.
* Value—the format of the attribute or element. Value may be an XML type (e.g., “string”) or a reference to another element description (e.g., “See Bar Element”). Annotations for limits or enumerations may be included (e.g.,” int [0..100]” to indicate an XML xs:int type with an accepted range from 1 to 100 inclusively)
* Card—cardinality of the element. If blank, then it is 1. Other typical values are 0..1 (optional), 1..n and 0..n.

The first row of the table after the header is the element being defined. This is immediately followed by attributes of this element, if any. Subsequent rows are child elements and their attributes. All child elements (i.e., those that are direct descendants) are included in the table. Simple child elements may be fully defined here (e.g., “Title”, “ ”, “Title of work”, “xs:string”), or described fully elsewhere (“POC”, “ ”, “Person to contact in case there is a problem”, “md:ContactInfo-type”). In this example, if POC was to be defined by a complex type defined as md:ContactInfo-type. Attributes immediately follow the containing element.

Accompanying the table is as much normative explanation as appropriate to fully define the element, and potentially examples for clarity. Examples and other informative descriptive text may follow. XML examples are included toward the end of the document and the referenced web sites.

### General Notes

All required elements and attributes must be included.

When enumerations are provided in the form ‘enumeration’, the quotation marks (‘’) should not be included.

UTF-8 [RFC3629] encoding shall be used when ISO/IEC 10646 (Universal Character Set) encoding is required.

## Normative References

[Avails] Content Availability Metadata, TR-META-AVAIL, <http://www.movielabs.com/md/avails>

[CM] Common Metadata, TR-META-CM, <http://www.movielabs.com/md/md>

[CMM] Common Media Manifest Metadata, TR-META-MMM, <http://www.movielabs.com/md/manifest>

[MEC] Media Entertainment Core, TR-META-MEC, , <http://www.movielabs.com/md/mec/>

[EIDR] Entertainment Identifier Registry (EIDR), <http://eidr.org/resources/>

[TR-META-CR] *Common Metadata Content Ratings*. [www.movielabs.com/md/ratings](http://www.movielabs.com/md/ratings). Note that a specific version is not referenced as it is intended that the latest version will be used. Referencing specifications may selection a specific version of the referenced document.

[TR-META-RS] Common Metadata Ratings Schema Definition, TR-META-RS, January 3, 2014, <http://www.movielabs.com/md/ratings/doc.html>

[XML] “XML Schema Part 1: Structures”, Henry S. Thompson, David Beech, Murray Maloney, Noah Mendelsohn, W3C Recommendation 28 October 2004, <http://www.w3.org/TR/xmlschema-1/> and “XML Schema Part 2: Datatypes”, Paul Biron and Ashok Malhotra, W3C Recommendation 28 October 2004, http://www.w3.org/TR/xmlschema-2/

## Informative References

## Best Practices for Maximum Compatibility

Metadata typically evolves with the addition of new elements, attributes and vocabularies. Existing applications should be capable of accepting metadata, even though there might be more data than expected. Strict XML validation precludes an orderly evolution and can be counterproductive to the flexibility needed in real implementations.

Metadata specifications and schema updates are designed to support backwards compatibility. For example, element and attributes can be added, but required elements are not removed; or more generally ordinality of elements and attributes can be widened but not narrowed. Values are not changed in either syntax or semantics. Therefore, we strongly encourage implementations to either be diligent in tracking to the latest version, or follow the backwards compatibility rules provided here.

An XML document is considered compatible if its structure does not preclude the extraction of data from the document. For example, a document with additional elements and attributes do not preclude schema parsing and data extraction.

* Do not reject compatible XML documents, unless they fail schema validation against the definition for an exact version/namespace match.
* Extract data from compatible XML documents whenever possible
* It is allowable to ignore elements and attributes whose presence is not allowed in the specification and schema versions against which the implementation was built. For example, if the original schema allows one instance and three instances are found, the 2nd and 3rd instance may be ignored.

We will try to update metadata definitions such that following these rules work consistently over time. Sometimes, changes must be made that are not always backwards compatible, so we will do our best to note these.

# Structure

## Requirements Structure

There are two parts to defining requirements: Scope (where the requirements apply) and Profiles (structured requirements).

### Scope

Scope defines where and when Profiles apply. Scope further divides into Territory and Category (TV, movies, etc.).

Territory is pretty straightforward. If the scope is worldwide, requirements apply everywhere, except where territory requirements are specified. This is an object model, where territories inherit the properties of the world, except where exceptions exist. There are specific rules that dictate what is inherited and what is not. {TBD}

Category defines what type of content, storefront, license model or other contextual parameter determines what rules apply. Like Region, Category is an object model where specifics inherit from their parent. For example, there could be a Category for TV and subcategory for Next-Day TV. Next-Day TV inherits most of its requirements (e.g., required artwork) from TV, but has different delivery timeframes.

Although inheritance can, at first, be daunting this is very much how people refer to content delivery requirements on paper.

### Profiles

A Profile describes requirements for some specific delivery. It takes several Profiles to fully describe a delivery.

Consider artwork for TV. It requires a collection of images with a particular aspect ratio and resolution; each with its own ‘purpose’. These are called Artwork Profiles. However, each image must comply with technical requirements such as encoding (JPEG, GIF, PNG), color encoding, maximum file size, and so forth. As all artwork images comply with a relatively small number of image specs, we have we have Image Profiles. Artwork Profiles simply refer to the applicable Image Profile.

Profiles come in the following categories

* Admin Profiles – Administrative rules such as lead times
* Language Profiles – Rules about localization, subs and dubs, and other language requirements as they apply to a territory
* Artwork Profiles – Sets of artwork, including resolutions, purpose, etc. [safe area?]
* Product Profiles – Definition of product-related deliverables, such as features, trailers, artwork, and bonus
* Technical Profiles – Audio, video, image, subtitle, and other digital asset technical descriptions

#### Product Profiles

A Product Profile defines requirements for Feature (main feature), Promotional (ads, such as trailers) and Supplemental (bonus/extras/VAM). Each of these can have their own content requirements covering technical requirements, artwork, metadata and parameters specific to the type.

One would generally expect to have distinct Product Profiles for moves and TV. One could additionally have Product Profiles for deep catalog or tentpole titles. For example, deep catalog might have relaxed technical requirements. Tentpole titles might have additional expectations on artwork, trailers (Promotional) or bonus (Supplemental).

[CHS: I think Admin Profile should be referenced from Product Profile, not from territory. Is that right?]

#### Admin Profiles

Admin profiles address logistics issues such as lead time and priority. This sets general rules about delivery.

[CHS: Product Profiles reference DeliveryParams. They should probably reference Admin Profiles. Key is to reference in the right place.]

#### Language Profiles

Language Profiles describe localization, including what artwork, metadata, audio, localized video, and other materials must be provided.

Language Profile is designed to provide defaults for information that would be found in EMA Avails [Avails]. Information in the Language Profile can be mapped directly to AllowedLanguages, AssetLanguage, LocalizationType, and RequiredFulfillmentLanguages.

#### Artwork Profiles

Each retail user interface has its own artwork requirements. Typically, there is a set of images for any given application. For example, movies might require 0.73 aspect ratio key art, while TV requires square key art. However, there can be more specific requirements, such as artwork for premium movies versus artwork for deep catalog movies.

Artwork Profiles are created for each set of images, each with a specific purpose (e.g., “cover1” or “hero2”). Purposes can correspond with MEC’s LocalizedInfo/ArtReference/@purpose, so when artwork is delivered you know exactly what you’re getting.

Image encoding (e.g., GIF/JPG/PNG, color space, etc.) is distinct from the Artwork Profile.

#### Technical Profiles

The following Technical Profiles are provided

* Audio
* Video
* Subtitle
* Image
* Cards
* Metadata
* Container

### Profile Examples

#### Technical Profile

The following illustrates potential Technical Profiles. These profiles are described rather than encoded in XML. Many details are omitted for brevity.

Following are example video profiles:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Profile Name** | **Codec** | **Aspect Ratio** | **Color Space** | **Primaries** | **Sub-sampling** | **Bit depth** | **Frame Rate** |
| HD ProRes | ProRes HQ | 4:3, 1.66:1, 16:9, 1.85:1, 2:1, 2.20:1, 2.35:1, 2.39:1, 2.40:1 | BT.709 | BT.709 | 4:2:2 or 4:2:2 | 8-bit or 10-bit | 23.976p, 24p, 25i, 25p, 29.97i, 29.97p, 30i, 30p, 60i |
| HD MPEG2 | MPEG-2 Main or High | 4:3, 1.66:1, 16:9, 1.85:1, 2:1, 2.20:1, 2.35:1, 2.39:1, 2.40:1 | BT.709 | BT.709 | 4:2:2 or 4:2:2 | 8-bit or 10-bit | 23.976p, 24p, 25i, 25p, 29.97i, 29.97p, 30i, 30p, 60i |
| HD AVC | H.264 Hight | 4:3, 1.66:1, 16:9, 1.85:1, 2:1, 2.20:1, 2.35:1, 2.39:1, 2.40:1 | BT.709 | BT.709 | 4:2:2 or 4:2:2 | 8-bit or 10-bit | 23.976p, 24p, 25i, 25p, 29.97i, 29.97p, 30i, 30p, 60i |
| UHD | ProRes | 4:3, 1.66:1, 16:9, 1.85:1, 2:1, 2.20:1, 2.35:1, 2.39:1, 2.40:1 | BT.709 | BT.709 | 4:2:2 | 10-bit | 23.976, 24, 25, 29.97, 30, 60 |
| UHDHDR | ProRes 422 HQ | 4:3, 1.66:1, 16.9, 1.85:1, 2:1, 2.20:1, 2.35:1, 2.39:1, 2.40:1 | BT.2100 | P3 | 4:2:2 | 10-bit | 23.976, 24, 25, 29.97, 30, 60 |

Following are example audio profiles

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Profile Name** | **Codec** | **Channel Layout** | **Sample Rate** | **Bit Depth** | **Min Bitrate** |
| PCM | PCM | ‘Mono’, ‘Mono, Mono’, ‘L,R’ ‘L,R,C,LFE,LS,RS’, ‘ ‘L,R,C,LFE,LS,RS,LRS,RRS’ | 48KHz | 16, 24 |  |
| MPL2-S | MPEG-2 Layer II | ‘Mono’, ‘Mono, Mono’, ‘L,R’ | 48KHz | 16, 24 | 384K |
| MPL2-MC | MPEG-2 Layer II | ‘L,R,C,LFE,LS,RS’, ‘ ‘L,R,C,LFE,LS,RS,LRS,RRS’ | 48KHz | 16, 24 | 912K |
| AC-3-S | AC-3 | ‘Mono’, ‘Mono, Mono’, ‘L,R’ | 48KHz | 16, 24 | 192K |
| AC-3-MC | AC-3 | ‘L,R,C,LFE,LS,RS’, ‘ ‘L,R,C,LFE,LS,RS,LRS,RRS’ | 48KHz | 16, 24 | 448K |
| AAC-S | AAC | ‘Mono’, ‘Mono, Mono’, ‘L,R’ | 48KHz | 16, 24 | 448K |
| AAC-MC | AAC | ‘L,R,C,LFE,LS,RS’, ‘ ‘L,R,C,LFE,LS,RS,LRS,RRS’ | 48KHz | 16, 24 | 960K |
| Atmos | EAC3-Atmos |  |  |  |  |

Given these Audio and Video Profiles, a Technical Profile might look like the following. It references the other profiles. Note that an actual package definition would also reference subtitles.

|  |  |  |  |
| --- | --- | --- | --- |
| **Profile Name** | **Container** | **Video Profiles** | **Audio Profiles** |
| MOV-HD | MOV | HD MPEG2, HD AVC | PCM, MPL2-S, MP2-MC, AC-3-S, AC-3-MC, AAC-S, AAC-MC |
| MOV-UHD | MOV | UHD, UHDHDR | PCM, MPL2-S, MP2-MC, AC-3-S, AC-3-MC, AAC-S, AAC-MC |
| ProRes-HD | ProRes | HD ProRes | PCM, AC-3-S, AC-3-MC, AAC-S, AAC-MC, Atmos |
| ProRes-UHD | ProRes | UHD, UHDHDR | PCM, AC-3-S, AC-3-MC, AAC-S, AAC-MC, Atmos |

[[CHS NOTE TO SELF: Can we define all this as a Container, then just reference Containers from a Package. OR, do we need to map individual objects into a container.

How about this:

* Technical Profiles are collected into a Container
* Product Profiles are collected into Packages
  + Profiles in Packages reference Containers (e.g., feature references a package, promotional references a package, etc.)
  + Profiles probably need to reference Artwork and Metadata
  + Package 🡪   
     Feature/Promotional/Supplemental   
     🡪 {Metadata + Artwork + AV} + {Tech Metadata + Tech Image + Tech Container} [Can also include Metadatat and Image in Container?] ]]

# General Types Encoding

## Simple Types

### RangeCondition

RangeCondition defines the range of acceptable technical parameters. RangeCondition is an xs:string and typically an attribute (@rangeCondition).

When values are expressed,

Acceptable values for @rangeCondition are as follows

* ‘min’ – Represents minimum requirement. If numeric, lower values are not accepted.
* ‘max’ – Represents the maximum acceptable value. If numeric, higher values are not accepted.
* ‘preferred’ – Represents preferred condition or value.
* ‘acceptable’ – Represents a condition or value that is acceptable but not desired. There may be negative consequences of using this condition, such as lower quality.

## DeliverySource-type

DeliverySource-type provides information about who provided the request. This can include both the Service Provider who generated the document as well as the retailer(s) for whom the document was prepared. This construct is useful to avoid ambiguity when requests come from service providers.

If requests are made on behalf of multiple retailers or storefronts, multiple Retail instances can be included.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Element** | **Attribute** | **Definition** | **Value** | **Card.** |
| **DeliverySource-type** |  |  |  |  |
| ServiceProvider |  | Service Provider delivering document | md:OrgName-type | 0..1 |
| Retailer |  | Retailer for whom the document was created | md:OrgName-type | 0..n |
| DeliveryContact |  | Contact information for this document, typically from a Service Provider. | md:ContactInfo-type | 0..1 |

## DeliveryHandling-type

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Element** | **Attribute** | **Definition** | **Value** | **Card.** |
| **DeliveryHandling-type** |  |  |  |  |
| Comments |  | Any comments. Should be included if ExceptionFlag=’true’ | xs:string | 0..1 |
| ExceptionFlag |  | Indicates message requires human attention | xs:boolean | 0..1 |
| Priority |  | Priority of request | TBD | 0..1 |
| ResponseDate |  | Expected response date | xs:date | 0..1 |
|  | dateIsTarget | If ‘true’ indicates ResponseDate is not a hard deadline. Details determined bilaterally. | xs:boolean | 0..1 |

## DeliveryInstructions-type

[CHS: I don’t particularly like this. Rename DeliveryHandling to DeliveryInstructions, fix links, and put OrderID where it’s needed.]

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Element** | **Attribute** | **Definition** | **Value** | **Card.** |
| **DeliveryInstructions-type** |  |  | delivery:DeliveryHandling-type |  |
| OrderID |  | Order identifier | md:id-type | 0.1 |

## Terms

Terms allows arbitrary terms to be specified.

The precise interpretation is subject to the mutual agreement of parties involved, although guidance is provided within.

Each term is a name/value pair with the name expressed as termName and the value expressed as one of Money, Event, Duration or text depending on the data contained within the term. If data cannot be otherwise expressed, the any##other element can be used.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Element** | **Attribute** | **Definition** | **Value** | **Card.** |
| **AvailTerms-type** |  |  |  |  |
|  | termName | Identifies the term. Enumeration is below. termName is case insensitive (i.e., case shall be ignored). | xs:string |  |
| Money |  | Used when termName refers to a term expressed in terms of money. | md:Money-type | (choice) |
| Event |  | Used when termName refers to a term expressed in terms of a date, or date and time. See Section **Error! Reference source not found.**. | xs:union(xs:date, xs:dateTime) |
| Duration |  | Used when termName refers to a term expressed in terms of a time duration. | xs:duration |
| Text |  | Used when a term can be expressed in text and it is not one of the other term types. | xs:string |
| Boolean |  | Used when term can be expressed as True or False | xs:boolean |
| URI |  | Used for URIs, including identifiers. | xs:anyURI |
| Language |  | Used for language. | xs:language |
| ID |  | Any identifier | md:id-type |
| YearDateTime |  | Year, date or date+time. For time-only use Time. | md:YearDateOrTime |
| Time |  | Time. May include time zone. | xs:time |
| Region |  | Geographic area | md:Region-type |
| <any> |  | Any other element. Used when a term cannot practically be expressed with one of the other element choices. | any ##other |

The Term specified is indicated by termName with the following conditions. Only one instance of each term may be included unless otherwise specified.

Following is a terms template.

|  |  |  |
| --- | --- | --- |
| termName | Interpretation | Element used |
|  |  | Text |
|  |  | Language |

## DeliveryParams-type

DeliveryParams-type includes delivery parameters that are common across media types, metadata, promotional, supplemental and other materials.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Element** | **Attribute** | **Definition** | **Value** | **Card.** |
| **DeliveryParams-type** |  |  |  |  |
| LeadTime |  | Lead time for deliverables relative to window start date. Negative values represent time before window. | xs:duration | 0..1 |
|  | durationIsTarget | If ‘true’ LeadTime is a target; that is, not a fixed duration | xs:boolean | 0..1 |
| Priority |  | Priority of delivery relative to other deliverables. | x:integer | 0..1 |
| AdditionalInstructions |  | Any additional instructions | xs:string | 0..1 |
| Terms |  | Any additional terms | delivery:Terms-type | 0..n |

LeadTime is expressed as a negative duration for deliverables that occur prior to the window (the typical case).

durationIsTarget indicate that LeadTime are aspirational. The degree to which this must be honored is subject to bilateral service level agreements.

[CHS: How do we encode Priority???]

### DeliveryResolution-type

Defines the resolution for an image or picture in pixels. If resolution specifies a minimum (i.e. @absolute = ‘false’ or is absent), aspect ratio of width and height is fixed. That is, they both must scale together to maintain aspect ratio.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Element** | **Attribute** | **Definition** | **Value** | **Card.** |
| **DeliveryResolution-type** |  |  |  |  |
| Width |  | Width in pixels | x:integer | 0..1 |
| Height |  | Height in pixels | x:integer | 0..1 |

## Types that reference objects directly

[CHS: Not relevant to CDR.]

### DeliveryObjectReference-type

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Element** | **Attribute** | **Definition** | **Value** | **Card.** |
| **DeliveryObjectReference-type** |  |  |  | 0..1 |
| TrackReference |  | TrackReference per [Manifest], Section 2.2.3 | xs:string | 0..n |
| TrackIdentifier |  | TrackIdentifier per [Manifest], Section 2.2.3 | md:ContentIdentifier-type | 0..n |
| EIDRURN |  | EIDR identifier along with structural type | delivery:EIDRURN | 0..n |
| TrackID |  | Reference track identifiers as per [Manifest] | delivery:DeliveryTrackID | 0..n |
| IMFRef |  | Reference to information in an Interoperable Master Format (IMF) file. | Delivery:DeliveryIMF-type | 0..n |

#### DeliveryTrackID-type

Allows tracks to be referenced

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Element** | **Attribute** | **Definition** | **Value** | **Card.** |
| **DeliveryTrackID-type** |  |  |  |  |
| AudioTrackID |  | Audio track ID | manifest:AudioTrackID-type | (choice) |
| VideoTrackID |  | Video track ID | manifest:VideoTrackID-type |
| SubtitleTrackID |  | SubtitleTrack ID | manifest:SubtitleTrackID-type |
| ImageID |  | Image ID | manifest:ImageTrackID-type |
| InteractiveTrackID |  | Interactive object (e.g., app) ID | manifest:InteractiveTrackID-type |
| ContentID |  | Content ID | md:ContentIID-type |
| AncillaryTrackID |  | Ancillary track ID | manifest:AncillaryTrackID-type |
| TextObjectID |  | Text object ID | manifest:TextObjectTrackID-type |

#### DeliveryIMFRef-type

References UUIDs for IMF CPLs, OPLs and virtual tracks.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Element** | **Attribute** | **Definition** | **Value** | **Card.** |
| **DeliveryIMFRef-type** |  |  | Extension of manifest:PresentationIMFRef-type |  |
| <TBD> |  |  |  |  |

### DeliveryFileReference-type

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Element** | **Attribute** | **Definition** | **Value** | **Card.** |
| **DeliveryFileReference-type** |  |  |  |  |
| FileInfo |  | Reference to a file. This file might be in a container. | manifest:FileInfo-type | 0..n |
| Container |  | Reference to container. | manifest:ContainerReference-type | 0..n |

## Types that reference objects through description

### DeliverObjectDescription-type

[CHS: This is very complete in that it can describe any instance. However, it’s not very good at describing ranges or options. More work is needed here.]

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Element** | **Attribute** | **Definition** | **Value** | **Card.** |
| **DeliveryObjectDescription-type** |  |  |  |  |
| Audio |  |  | manifest:InventoryAudio-type | 0..n |
| Video |  |  | manifest:InventoryVideo-type | 0..n |
| Subtitle |  |  | manifest:InventorySubtitle-type | 0..n |
| Image |  |  | manifest:InventoryImage-type | 0..n |
| Interactive |  |  | manifest:InventoryInteractive-type | 0..n |
| Ancillary |  |  | manifest:InventoryAncillary-type | 0..n |
| Metadata |  |  | manifest:InventoryMetadata-type | 0..n |
| TextObject |  |  | manifest:InventoryTextObject-type | 0..n |
| ExternalManifest |  |  | manifest:InventoryManifest-type | 0..n |

# Profiles

A Profile is a collection of requirements. Currently, we refer to

* Administrative Profile – Lead times, priorities, and special instructions
* Language Profiles – Sets of timed text and audio (i.e., subs and dubs) rules by language
* Technical Profiles – Technical requirements about tracks

Once defined, a Profile is used as shorthand for these requirements. For example, one might have a “Benelux” profile for language requirements for Benelux countries, and an “HDR” profile for minimum HDR requirements.

Profiles can be referenced both as requirements and as part of deliveries. That is, a Content Delivery Requirements (CDR) document might define an “HDR” profile, an MMC delivery might refer to the assets as fulfilling part of the “HDR” Profile; and, an Asset Status Manifest might indicate the “HDR” Profile has not yet been delivered.

## Administrative Profile

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Element** | **Attribute** | **Definition** | **Value** | **Card.** |
| **DeliveryAdminProfile-type** |  |  |  |  |
| AdminProfileID |  | ID for this profile | xs:string |  |
| LeadTime |  | Lead time for deliverables relative to window start date. Negative values represent time before window. | xs:duration | 0..1 |
|  | durationIsTarget | If ‘true’ LeadTime is a target; that is, not a fixed duration | xs:boolean | 0..1 |
| Priority |  | Priority of delivery relative to other deliverables. | x:integer | 0..1 |
| AdditionalInstructions |  | Any additional instructions | xs:string | 0..1 |
| Terms |  | Any additional terms | delivery:Terms-type | 0..n |

## Language Profiles

[CHS: Need to grab more of the Language information from XML Avails]

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Element** | **Attribute** | **Definition** | **Value** | **Card.** |
| **DeliveryLanguageProfile-type** |  |  |  |  |
| LangProfileID |  | ID for this profile | xs:string |  |
| Language |  | Language for which this Profile was not defined | xs:language | 0..n |
| ExcludedLanguage |  | Excluded Languages for which this Profile was not defined | xs:language | 0..n |
| LocalizationType |  | Localization Type, using encoding from [Avails] | xs:string | 0..1 |
| DescriptiveAudio |  | Is descriptive audio required. ‘true’ means yes. | xs:boolean | 0..1 |
| Term |  | Additional terms that apply to this Profile | delivery:Terms-type | 0..n |

For Language and ExcludedLanguage, use semantics as defined in Media Manifest [Manifest].

## Artwork Profiles

This type defines a profile images each of which constitute artwork serving a ‘purpose’. Typically, that purpose, defined in @purpose, corresponds with [CM] LocalizedInfo/ArtReference/@purpose.

An instance is included for each combination of @purpose and @imageProfileName. If @imageProfileName is absent, the default Image Profile is used. If there is only one TechImage-type/ImageProfile, it is the default. ImageProfile/@default = ‘true’, it is the default.

### ArtworkProfile-type

When multiple instances of Aspect or Resolution are provided, each of those is required. Aspect should not be included for the same image.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Element** | **Attribute** | **Definition** | **Value** | **Card.** |
| **ArtworkProfile-type** |  | Base type for this element is standard delivery parameters defined in DeliveryParams-type. | delivery:DeliveryParams-type (by extension) |  |
|  | artworkProfileID | Image profile name corresponding with ImageProfile in DeliveryImage-type | md:id-type | 0..1 |
| Image |  | Image with a given purpose that is part of this profile | Delivery:ArtworkProfileImage-type | 1..n |

### ArtworkImage-type

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Element** | **Attribute** | **Definition** | **Value** | **Card.** | |
| **ArtworkImage-type** |  |  |  |  | |
|  | purpose | Image purpose | xs:string | 0..1 | |
|  | imageProfileName | Unique image name. Note that @purpose could appear in multiple profiles. | xs:string | 0..1 | |
| ImageAspectRatio |  | Aspect ratio represented as a decimal number representing the ratio between the x-axis and y-axis dimensions. Note this definition is distinct from [CM] Picture/AspectRatio which is a string. | xs:decimal | 1..n | choice |
|  | rangeCondition | Range Condition in accordance with Section 3.1.1. | xs:string | 0..1 |
| Resolution |  | Resolution of image (fixed or minimum) | delivery:DeliveryResolution-type | 1..n |
|  | rangeCondition | Range Condition in accordance with Section 3.1.1. | xs:string | 0..1 |

## Technical Profiles

The Technical Profiles is a collection of audio, video, subtitle, dub card, image, metadata and container profiles. Each component profile is defined independently so it can be reused across Technical Profiles.

The TechnicalProfiles-type defines a set of Technical Profiles (TechProfile). It also defines the component profiles (Audio, Video, etc.).

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Element** | **Attribute** | **Definition** | **Value** | **Card.** |
| **DeliveryTechnicalProfiles-type** |  |  |  |  |
| TechProfile |  | A Technical Profile. | delivery:TechnicalProfile-type | 1..n |
| Audio |  | Parameters than define acceptable audio media delivery. | delivery:DeliveryAudio-type | 0..1 |
| Video |  | Parameters than define acceptable video media delivery. | delivery:DeliveryVideo-type | 0..1 |
| Subtitle |  | Parameters than define acceptable timed text media delivery. | delivery:DeliverySubtitle-type | 0..1 |
| Metadata |  | Parameters than define acceptable metadata delivery. | delivery:DeliveryMetadata-type | 0..1 |
| Image |  | Parameters that define acceptable image delivery, including artwork | delivery:DeliveryImage-type | 0..1 |
| Promotional |  | Parameters than define acceptable promitional material delivery. | delivery:DeliverySupplemental-type | 0..1 |
| Supplemental |  | Parameters than define acceptable supplemental material delivery. | delivery:DeliverySupplemental-type | 0..1 |
| Term |  | Additional terms that apply to this Profile | delivery:Terms-type | 0..n |

### TechnicalProfile-type

TechProfile-type defines a single Technical Profile.

Technical Profiles specify acceptable parameters for media of their respective types. Constraints consist of an XML elements corresponding with characteristics of the media.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Element** | **Attribute** | **Definition** | **Value** | **Card.** |
| **DeliveryTechnicalProfile-type** |  |  |  |  |
| TechProfile | @techProfileID | Unique identifier for this Technical Profile | md:id-type | 1..n |
| Audio |  | Parameters than define acceptable audio media delivery. | delivery:TechAudio-type | 0..n |
| Video |  | Parameters than define acceptable video media delivery. | delivery:TechVideo-type | 0..n |
| Subtitle |  | Parameters than define acceptable timed text media delivery. | delivery:TechSubtitle-type | 0..n |
| Metadata |  | Parameters than define acceptable metadata delivery. | delivery:TechMetadata-type | 0..n |
| Image |  | Parameters that define acceptable image delivery, including artwork | delivery:TechImage-type | 0..n |
| Promotional |  | Parameters than define acceptable promotional material delivery. | delivery:TechSupplemental-type | 0..n |
| Supplemental |  | Parameters than define acceptable supplemental material delivery. | delivery:TechSupplemental-type | 0..n |
| Term |  | Additional terms that apply to this Profile | delivery:Terms-type | 0..n |

### Interpretation of terms within Technical Profiles

Technical Profiles contain parameters that correspond with technical characteristics of media files. Most of these correspond with technical values in Common Metadata [CM]. The full definitions are found in the referenced sections of Common Metadata.

When Type includes the note “Incl. @rangeCondition” then the type as defined in [CM] is extended to include an @rangeCondition attribute.

When a term is absent, there are no constraints. For example, if MaxFileSize is not specified, there are no limits on size. If Compliance is absent, there are no additional Compliance constraints. Generally speaking, only constrained parameters should be included. This makes the profile shorter and less complicated.

When a term is present, interpretation depends on the value of @rangeCondition.

When @rangeCondition is ‘preferred’, that is a suggestion, not a hard requirement.

For numeric values, elements values can be provided with @rangeCondition of ‘min’ and/or ‘max’. Values are inclusive. It is allowed to specify either or both of ‘min’ or ‘max’. With both are specified, media characteristic must fall within that limit (inclusive). When only a ‘min’ value is included, there is a fixed minimum but no maximum. With only a ‘max’ value is provided there is a fixed maximum with no minimum. There can be at most one ‘min’ value and one ‘max’ value.

For numeric values, any value with @rangeCondition of ‘preferred’ must be <= a ‘max’ value and >= a ‘min’ value. ‘min’ values must be <= ‘max’ values. At most one ‘min’ and one ‘max’ may be included. There is no limit on ‘preferred’ values. For example, 48kHz and 44.1kHz may both be ‘preferred’ values.

Non-numeric values may not have @rangeCondition = ‘max’ or ‘min’. This might be tempting for values such coded profiles, but it can sometimes be ambiguous.

### TechAudio-type

References to Common Metadata types in this section refer to object in DigitalAssetImageData-type, as defined in [CM] section 5.2.3, with the same name.

.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Element** | **Attribute** | **Definition** | **Value** | **Card.** |
| **TechAudio-type** |  |  |  |  |
|  | audioTechProfileName | Unique name of technical profile. If there is only one profile of this type and @default=‘true’, this need not be included. | md:id-type | 0..1 |
|  | Default | This profile is the default profile. If ‘true’, it is. If absent or ‘false’ it is not default. At most one instance can be the default | xs:boolean | 0..1 |
| Codec |  | As defined in [CM] | Incl. @rangeCondition | 0..1 |
| CodecType |  | As defined in [CM] | Incl. @rangeCondition | 0..n |
| BitrateMax |  | As defined in [CM] | Incl. @rangeCondition | 0..n |
| VBR |  | As defined in [CM]. | Incl. @rangeCondition | 0..1 |
| SampleRate |  | As defined in [CM] | Incl. @rangeCondition | 0..n |
| SampleBitDepth |  | As defined in [CM] | Incl. @rangeCondition | 0.n |
| Channels |  | As defined in [CM] | Incl. @rangeCondition | 0.n |
| ChannelMapping |  | As defined in [CM] | Incl. @rangeCondition | 0.n |
| Compliance |  | As defined in [CM] | Incl. @rangeCondition | 0.n |
| Loudness |  | As defined in [CM] | Incl. @rangeCondition | 0..1 |
| MaxFileSize |  | Maximum file size in bytes for file of this type | xs:nonNegativeInteger | 0..1 |
| Term |  | Additional terms that apply to this Profile | delivery:Terms-type | 0..n |

### TechVideo-type

References to Common Metadata types in this section refer to object in DigitalAssetVideoData-type, as defined in [CM] section 5.2.4, with the same name.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Element** | **Attribute** | **Definition** | **Value** | **Card.** |
| **TechVideo-type** |  |  |  |  |
|  | videoTechProfileName | Unique name of technical profile. If there is only one profile of this type and @default=‘true’, this need not be included. | md:id-type | 0..1 |
|  | Default | Is this the default profile. If ‘true’, it is. If absent or ‘false’ it is not default. At most one instance can be the default | xs:boolean | 0..1 |
| FrameCharacteristics |  | Frame constraints |  |  |
| ColorCharacteristics |  | Color constraints |  |  |
| NextGenCharacteristics |  | Next Gen (i.e., HDR) characteristics |  |  |
| Type3D |  | As defined in [CM] |  |  |
| MasterText |  | Defines the text allowed in the master | xs:string | 0..1 |
|  | titles | Title text allowed. ‘true’ means allowed | xs:boolean | 0..1 |
|  | credits | Credit text allowed. ‘true’ means allowed | xs:boolean | 0..1 |
|  | scene | Scene setting text allowed, ‘true’ means allowed | xs:boolean | 0..1 |
|  | forced | Force narrative text allowed. ‘true’ means allowed | xs:boolean | 0..1 |
|  | textlessElements | Textless elements (i.e., video without text) provided in conjunction with texted video. ‘true’ means provided | xs:boolean | 0..1 |
| DiscreteCards |  | Indicates cards are delivered separately from video. If only certain cards are provided discretely, attributes indicate which ones are discrete. If cards are not discrete, they are appended to video and are part of the timeline. | xs:boolean | 0..1 |
|  | dub | Dub cards are discrete | xs:boolean | 0..1 |
|  | rating | Rating cards are discrete | xs:boolean | 0..1 |
|  | territory | Territory-specific cards, such as anti-piracy and health cards, are discrete | xs:boolean | 0..1 |
| Compliance |  | As defined in [CM] | Incl. @rangeCondition | 0.n |
| MaxFileSize |  | Maximum file size in bytes for file of this type | xs:nonNegativeInteger | 0..1 |
| Term |  | Additional terms that apply to this Profile | delivery:Terms-type | 0..n |

There are many definitions of terms like “semi-textless” based on what texted elements are allowed. The intent is to define what text elements are allowed in the video or need to be removed. For this purpose, we define text in terms of the following

* Titles – opening and closing
* Credits – opening and closing credits
* Scene Setting – Scene setting text such as location or time
* Forced narrative – Forced subtitles
* Photographic – Any text captured in a scene during production, such as billboards and street signs. Does not include VFX or animation-produced text. Production text is assumed to be part of the video, and is not considered in the context of texted or textless masters.

MasterText is encoded as follows. Note that most profiles prefer texted and/or semi-textless masters.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **MasterText** | **Titles** | **Credits** | **Scene Setting** | **Forced Narrative** | **Additional** |
| ‘Texted’ | Allowed | Allowed | Allowed | Allowed |  |
| ‘Semi-textless’ | Allowed | Allowed | Allowed | Prohibited |  |
| ‘Textless’ | Prohibited | Prohibited | Prohibited | Prohibited |  |
| ‘TextlessElements’ | Allowed | Allowed | Allowed | Allowed | Textless elements are provided with texted master, typically appended |
| ‘Other’ |  |  |  |  | Allowed text defined in attributes. |

#### TechVideoFrame-type

References to Common Metadata types in this section refer to object in DigitalAssetVideoPicture-type, as defined in [CM] section 5.2.6, with the same name.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Element** | **Attribute** | **Definition** | **Value** | **Card.** |
| **TechVideoFrame-type** |  |  |  |  |
| Resolution |  | As defined in [CM] | Incl. @rangeCondition | 0..n |
| AspectRatio |  | As defined in [CM] | Incl. @rangeCondition | 0..n |
| PixelAspect |  | As defined in [CM] | Incl. @rangeCondition | 0..n |
| FrameRate |  | As defined in [CM] | Incl. @rangeCondition | 0..n |
| Progressive |  | As defined in [CM] | Incl. @rangeCondition | 0..n |
| LetterboxAccepted |  | Letterbox and Pillarbox video is accepted. If ‘false’, only active pixels should be provided. | xs:boolean | 0..1 |

#### TechVideoColor-type

References to Common Metadata types in this section refer to object in DigitalAssetVideoPicture-type, as defined in [CM] section 5.2.6, with the same name.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Element** | **Attribute** | **Definition** | **Value** | **Card.** |
| **TechVideoColor-type** |  |  |  |  |
| ColorSubsampling |  | As defined in [CM] | Incl. @rangeCondition | 0..n |
| MasteredColorVolume |  | As defined in [CM] | Incl. @rangeCondition | 0..n |
| BitDepth |  | As defined in [CM] | Incl. @rangeCondition | 0..n |

#### TechVideoNextGen-type

References to Common Metadata types in this section refer to object in DigitalAssetVideoPicture-type, as defined in [CM] section 5.2.6, with the same name.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Element** | **Attribute** | **Definition** | **Value** | **Card.** |
| **TechVideoNextGen-type** |  |  |  |  |
| LightLevel |  | As defined in [CM] | Incl. @rangeCondition | 0..n |
| ColorVolumeTransform |  | As defined in [CM] | Incl. @rangeCondition | 0..n |

### TechSubtitle-type

References to Common Metadata types in this section refer to object in DigitalAssetSubtitleData-type, as defined in [CM] section 5.2.7, with the same name.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Element** | **Attribute** | **Definition** | **Value** | **Card.** |
| **TechSubtitle-type** |  |  |  |  |
|  | subtitleTechProfileName | Unique name of technical profile. If there is only one profile of this type and @default=‘true’, this need not be included. | md:id-type | 0..1 |
|  | Default | Is this the default profile. If ‘true’, it is. If absent or ‘false’ it is not default. At most one instance can be the default | xs:boolean | 0..1 |
| Type |  | As defined in [CM] | Incl. @rangeCondition | 0..n |
| Format |  | As defined in [CM] | Incl. @rangeCondition | 0..n |
| FormatType |  | As defined in [CM] | Incl. @rangeCondition | 0..n |
| Compliance |  | As defined in [CM] | Incl. @rangeCondition | 0..n |
| MaxFileSize |  | Maximum file size in bytes for file of this type | xs:nonNegativeInteger | 0..1 |
| Term |  | Additional terms that apply to this Profile | delivery:Terms-type | 0..n |

### TechCard-type

Technical description for card, such as dub cards.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Element** | **Attribute** | **Definition** | **Value** | **Card.** |
| **TechCard-type** |  |  |  |  |
|  | cardTechProfileName | Unique name of technical profile. If there is only one profile of this type and @default=‘true’, this need not be included. | md:id-type | 0..1 |
|  | Default | Is this the default profile. If ‘true’, it is. If absent or ‘false’ it is not default. At most one instance can be the default | xs:boolean | 0..1 |
| DiscreteCards |  | Indicates whether Discrete Cards are required | xs:boolean | 0..1 |
|  | rangeCondition | Range Condition in accordance with Section 3.1.1. | xs:string | 0..1 |
| MustMatchVideoEncoding |  | Indicates whether cards must match video encoding | xs:boolean | 0..1 |
| MustMatchVideoDynamicRange |  | Indicates whether cards must match video dynamic range. For example, if video is HDR, must the cards be HDR. | xs:boolean | 0..1 |
| MaxFileSize |  | Maximum file size in bytes for file of this type | xs:nonNegativeInteger | 0..1 |
| Term |  | Additional terms that apply to this Profile | delivery:Terms-type | 0..n |

### DeliveryImage-type

This object defines image technical characteristics. A set of image characteristics is called an Image Profile.

References to Common Metadata types in this section refer to object in DigitalAssetImageData-type, as defined in [CM] section 5.2.8, with the same name. Pixels are assumed to be square.

The image profile may be given a name in @imageProfileName. If this name is absent, it is assumed that all images will conform to this profile. Otherwise, artwork definitions must reference a named profile.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Element** | **Attribute** | **Definition** | **Value** | **Card.** |
| **DeliveryImage-type** |  | Base type for this element is standard delivery parameters defined in DeliveryParams-type. | delivery:DeliveryParams-type (by extension) |  |
|  | imageTechProfileName | Unique name of technical profile. If there is only one profile of this type and @default=‘true’, this need not be included. | md:id-type | 0..1 |
|  | Default | Is this the default profile. If ‘true’, it is. If absent or ‘false’ it is not default. At most one instance can be the default | xs:boolean | 0..1 |
|  | imageProfileName | Name of this image profile. | xs:string | 0..1 |
|  | default | This image profile is the default profile to be used when there is no reference to imageProfileName. At most on image profile may have @default = ‘true’ | xs:boolean | 0..1 |
| Encoding |  | As per Common Metadata definition. One for each acceptable encoding method. | xs:string | 0..n |
| AlphaAllowed |  | Is alpha channel supported (i.e., transparency). ‘true’ means yes. This must be absent or ‘false’ for encoding types that do not support alpha. | xs:boolean | 0..1 |
| DynamicRangeProfile |  | As defined in [CM] | xs:string | 0..1 |
| ColorGamutProfile |  | As defined in [CM] | xs:string | 0..1 |
| Compliance |  | As defined in [CM] | md:Compliance-type | 0..1 |
| MaxFileSize |  | Maximum file size in bytes for file of this type | xs:nonNegativeInteger | 0..1 |
| Term |  | Additional terms that apply to this Profile | delivery:Terms-type | 0..n |

### TechMetadata-type

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Element** | **Attribute** | **Definition** | | **Value** | **Card.** |
| **TechContainer-type** |  | Base type for this element is standard delivery parameters defined in DeliveryParams-type. | | delivery:DeliveryParams-type (by extension) |  |
|  | metadataTechProfileName | Unique name of technical profile. If there is only one profile of this type and @default=’true’, this need not be included. | md:id-type | | 0..1 |
|  | Default | Is this the default profile. If ‘true’, it is. If absent or ‘false’ it is not default. At most one instance can be the default | xs:boolean | | 0..1 |
| Encoding |  | What is the metadata schema. | | xs:string | 1..n |
|  | minVersion | Minimum version | | xs:string | 0..1 |
|  | maxVersion | Maximum version | |  |  |
|  | rangeCondition | Range Condition in accordance with Section 3.1.1. | | xs:string | 0..1 |
| MaxFileSize |  | Maximum file size in bytes for file of this type | | xs:nonNegativeInteger | 0..1 |
| Term |  | Additional terms that apply to this Profile | | delivery:Terms-type | 0..n |

Encoding is can be encoded with any value recognized by the recipient. However, Media Entertainment Core must be encoded as ‘MEC’. minVersion and maxVersion indicate the version of that metadata type. For example, if any version of MEC 2.5 and beyond is acceptable, minVersion should be ‘2.5’.

### TechContainer-type

References to Common Metadata types in this section refer to object in ContainerMetadataData-type, as defined in [CM] section 6.2, with the same name.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Element** | **Attribute** | **Definition** | | **Value** | **Card.** |
| **TechContainer-type** |  | Base type for this element is standard delivery parameters defined in DeliveryParams-type. | | delivery:DeliveryParams-type (by extension) |  |
|  | containerTechProfileName | Unique name of technical profile. If there is only one profile of this type and @default=‘true’, this need not be included. | md:id-type | | 0..1 |
|  | Default | Is this the default profile. If ‘true’, it is. If absent or ‘false’ it is not default. At most one instance can be the default | xs:boolean | | 0..1 |
| ContainerType |  | As defined in [CM] | | Incl. @rangeCondition | 0..n |
| MaxFileSize |  | Maximum file size in bytes for file of this type | | xs:nonNegativeInteger | 0..1 |
| Term |  | Additional terms that apply to this Profile | | delivery:Terms-type | 0..n |

## Product Profiles

ProductProfiles-type is the collection of Product Profiles

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Element** | **Attribute** | **Definition** | **Value** | **Card.** |
| **ProductProfiles-type** |  | Base type for this element is standard delivery parameters defined in DeliveryParams-type. | delivery:DeliveryParams-type (by extension) |  |
| ProductProfile |  | Product Profile definition | delivery:ProfileProfile-type | 0..1 |

### ProductProfile-type

This type defines a single Product Profile

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Element** | **Attribute** | **Definition** | **Value** | | **Card.** |
| **ProductProfile-type** |  |  |  | |  |
|  | productProfileID | Unique identifier for this Product Profile | md:id-type | | 0..1 |
|  | Default | Indicates whether this the default profile. If ‘true’, it is. If absent or ‘false’ it is not default. At most one instance can be the default | xs:boolean | 0..1 | |
| Feature |  | Feature characteristics | delivery:ProductProfile-type | | 0..n |
| Promotional |  | Promotional material characteristics | delivery:ProductPromotional-type | | 0..n |
| Supplemental |  | Supplemental material characteristics | delivery:ProductSupplemental-type | | 0..n |

### ProductProfileInfo-type

This type is the base type for Product Profiles. It contains data that is in all Product Profiles.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Element** | **Attribute** | **Definition** | **Value** | **Card.** |
| **ProductProfile-type** |  |  |  |  |
| TechProfileName |  | Name of Technical Profiles that apply to this Product Profile | xs:string | 0..n |
| ArtworkProfileName |  | Name of Artwork Profiles that apply to this Product Profile | xs:string | 0..n |
| LocalizedMetadata |  | Is localized metadata required for this Profile [CHS: not sure this belongs here.] | xs:boolean | 0..n |

### DeliveryPromotional-type

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Element** | **Attribute** | **Definition** | **Value** | **Card.** |
| **ProductPromotional-type** |  | Base type for this element is default Product Profile data | delivery:ProductProfileInfo-type (by extension) |  |
| IncludesTrailer |  | Indicates whether trailer is expected. ‘true’ means trailer is expected. | xs:boolean | 0..1 |
| LimitedAudience |  | Indicates limited audience promotional material is allowed (e.g., Red Band trailers) | xs:boolean | 0..1 |

### DeliverySupplemental-type

Supplementary material is any audiovisual, gallery, game, app, or other content that supplements the feature. Also referred to as Bonus and VAM (value added material).

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Element** | **Attribute** | **Definition** | **Value** | **Card.** |
| **DeliverySupplemental-type** |  | Base type for this element is default Product Profile data | delivery:ProductProfileInfo-type (by extension) |  |
| LocalizedBonus |  | Indicates whether supplemental material is expected to be localized to the territory. ‘true’ means supplemental material should be localized. | xs:boolean | 0..1 |
| IncludesBonus |  | Indicates whether supplemental material is expected. ‘true’ means supplemental material is expected. | xs:boolean | 0..1 |

# Content Delivery Requirements

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Element** | **Attribute** | **Definition** | **Value** | **Card.** |
| **DeliveryRequirements-type** |  |  |  |  |
|  | udpateNum, workflow, updateDeliveryType, versionDescription | Common set of workflow attributes (defined in Common Metadata) | md:Workflow-attr |  |
| Compatibility |  | Spec compatibility | manifest:Compatibility-type |  |
| Source |  | Source of CRD | delivery:DeliverySource-type | 0..1 |
| CDRID |  | Identifier for set of content delivery rules | md:id-type | 0..1 |
| Description |  | Description of content delivery rules set. | xs:string | 0..1 |
| Publisher |  | Content provider who will fulfill content in accordance with these content delivery rules | md:OrgName-type | 0..1 |
| GeneralRules |  | General and worldwide instructions | delivery:GeneralRules-type | 0..1 |
| TerritoryRules |  | Territory-specific rules | delivery:TerritoryRules-type | 0..1 |
| GeneralTerm |  | Additional terms, not covered by GeneralRules | delivery:Terms-type | 0..1 |
| Instructions |  | Handling instructions. Includes exception flag. | delivery:Instructions-type | 0..1 |

## GeneralRules-type

General rules apply across all territories, except when covered in territory rules—territory rules, including Category rules, take precedence. For example, if there is general rule about delivery times in GeneralRules, that applies everywhere, except where there are rules in specific TerritoryRules elements that cover the same delivery time parameters.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Element** | **Attribute** | **Definition** | **Value** | **Card.** |
| **DeliveryGeneralRules-type** |  |  |  |  |
| AdminProfileID |  | Reference to applicable Administrative Profile | md:id-type | 0..1 |
| LangProfileID |  | Reference to applicable Language Profile | md:id-type | 0..1 |
| TechProfileID |  | Reference to applicable Technical Profile | md:id-type | 0..1 |
| Terms |  | Additional terms | delivery:Terms-type | 0..1 |

## TerritoryRules-type

Territory rules apply across all categories within the territory, except when covered in category rules—category rules take precedence.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Element** | **Attribute** | **Definition** | **Value** | **Card.** |
| **DeliveryTerritoryRules-type** |  |  |  |  |
| Region |  | Region and Excluded Region define the territories where rues apply. They are encoded in accordance with Media Manifest [Manifest] Region and ExcludedRegion. | md:Region-type | (choice) |
| ExcludedRegion |  | md:Region-type |
| CategoryRules |  | Category-specific rules | Delivery:CategoryRules-type | 0..1 |
| AdminProfileID |  | Reference to applicable Administrative Profile | md:id-type | 0..1 |
| LangProfileID |  | Reference to applicable Language Profile | md:id-type | 0..1 |
| TechProfileID |  | Reference to applicable Technical Profile | md:id-type | 0..1 |
| Terms |  | Additional terms | delivery:Terms-type | 0..1 |

### CategoryRules-type

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Element** | **Attribute** | **Definition** | **Value** | **Card.** |
| **DeliveryCategoryRules-type** |  |  |  |  |
| ContentyCategory |  | Content Category for rules defined in this object. | xs:string |  |
| ContentSubCategory |  | Additional specificity of Content Category for rules defined in this object. | xs:string | 0..1 |
| AdminProfileID |  | Reference to applicable Administrative Profile | md:id-type | 0..1 |
| LangProfileID |  | Reference to applicable Language Profile | md:id-type | 0..1 |
| TechProfileID |  | Reference to applicable Technical Profile | md:id-type | 0..1 |
| Term |  | Additional terms that apply to this category and sub-category. | delivery:Terms-type | 0..n |

ContentCategory and ContentSubCategory define the scope of the CategoryRules object. When ContentDeliveryRequirements are used in conjunction with EMA Avails, ContentCategory values should correspond with Avails WorkType values. That allows an unambiguous linkage to Avails. ContentSubCategory can include values of WorkTypeDetail, values of EMA Avails LicenseTypeDescription (e.g., “Next Day TV” or “POD”), or other values that define handling (e.g., “Priority” and “Library”).

# Asset Order

## DeliveryRequest-type

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Element** | **Attribute** | **Definition** | **Value** | **Card.** |
| **DeliveryRequest-type** |  |  |  |  |
|  | updateNum, workflow, updateDeliveryType, versionDescription | Workflow attributes | md:Worflow-attr | 0..1 |
| DeliveryID |  |  | md:id-type | 0..1 |
| Description |  | Description of request | xs:string | 0..1 |
| Source |  | Source of this request | delivery:DeliverySource-type |  |
| Publisher |  | Publisher that originated content (i.e., generated the Avail) | md:orgName-type | 0..1 |
| ALID |  | ALID of content | md:id-type | 0..1 |
| FileReference |  | Reference to files requested | delivery:DeliveryFileReference-type | 0..n |
| ObjectReference |  | Reference to objects, such as specific tracks, requested | delivery:DeliveryObjectReference-type | 0..n |
| ObjectDescription |  | Reference to objects, such as tracks, by description (e.g., *French dub*). | delivery:DeliveryObjectDesription-type | 0..n |
| Instructions |  | Any other instructions | xs:string | 0..1 |

# Asset Status Manifest (ASM)

TBD

# QC Report

## QCReport-type

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Element** | **Attribute** | **Definition** | **Value** | **Card.** |
| **QCError-type** |  |  |  |  |
| ErrorDescription |  | Description of the issue with the media and/or file | delivery:ErrorDescription-type | 1..n |
| MediaAsset |  | Media Asset that is the subject of the error | delivery:ObjectReference-type | 0..n |
| FileReference |  | File that is the subject of the error | delivery:FileReference-type | 0..n |

## Error types

### QCError-type

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Element** | **Attribute** | **Definition** | **Value** | **Card.** |
| **QCError-type** |  |  |  |  |
|  | updateNum, workflow, updateDeliveryType, versionDescription | Workflow attributes | md:Worflow-attr | 0..1 |
| ErrorDescription |  | Description of the issue with the media and/or file | delivery:QCErrorDescription-type | 1..n |
| MediaReference |  | Media Asset that is the subject of the error | delivery:DeliveyrObjectReference-type | 0..n |
| FileReference |  | File that is the subject of the error | delivery:DeliveryFileReference-type | 0..n |

### QCErrorDescription-type

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Element** | **Attribute** | **Definition** | **Value** | **Card.** |
| **QCErrorDescription-type-type** |  |  |  |  |
| ErrorCategory |  | Error Category, in accordance with QC Nomenclature [ref] | xs:string |  |
| ErrorTerm |  | Error Term in accordance with QC Nomenclature [ref] | xs:string |  |
| CategorySpecific |  | Additonal data associated with error, based on Error Category. | delivery:QCCategoryError-type | 0..n |
| Comments |  | Any additional comments | xs:string | 0..1 |
| FullOrPartialQC |  | Indicates whether assets was fully evaluated or if evaluation stopped at first error(s) | xs:string | 0..1 |

FullOrPartialQC is encoded as follows [CHS: Is this just a boolean?]

* ‘Full’ – QC was completed
* ‘Partial’ – QC was aborted once error(s) were found. Additional errors may be present.

### QCCategoryError-type

This section contains additional information for errors. Value depends on the QC Nomenclature Category of the error.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Element** | **Attribute** | **Definition** | **Value** | **Card.** |
| **DeliveryCategoryError-type** |  |  |  |  |
| Audio |  | Audio Category error specifics | delivery:QCErrorAudio-type | (choice) |
| Video |  | Video Category error specifics | delivery:QCErrorVideo-type |
| TimedText |  | TimedText Category error specifics | delivery:QCErrorSubtitle-type |
| Metadata |  | Metadata Category error specifics | delivery:QCErrorMetadata-type |
| Artwork |  | Artwork Category error specifics | delivery:QCErrorArtwork-type |
| Package |  | Package Category error specifics | delivery:QCErrorPackage-type |

[CHS: Everything following is very preliminary.]

#### QCErrorAudio-type

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Element** | **Attribute** | **Definition** | **Value** | **Card.** |
| **QCErrorAudio-type** |  |  |  |  |
| StartTimecode |  | Track timeline where issue starts. Omit if issue exists for entire period or if start is unknown | manifest:Timecode-type | 0..1 |
| EndTimecode |  | Track timeline where issue ends. Omit, if problem persists to end of timeline or if end is unknown | manifest:Timecode-type | 0..1 |
| TimeOffset |  | For errors with alignment issues (e.g., AV Sync), the duration of offset. Negative means audio is ahead of video. | xs:duration | 0..1 |

#### QCErrorVideo-type

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Element** | **Attribute** | **Definition** | **Value** | **Card.** |
| **QCErrorVideo-type** |  |  |  |  |
| StartTimecode |  | Track timeline where issue starts. Omit if issue exists for entire period or if start is unknown | manifest:Timecode-type | 0..1 |
| EndTimecode |  | Track timeline where issue ends. Omit, if problem persists to end of timeline or if end is unknown | manifest:Timecode-type | 0..1 |
| XOffset |  | In pixels, x-value of lower left corner of issue. Omit if issue covers entire picture. | xs:decimal | 0..1 |
| YOffset |  | In pixels, y-value of lower left corner of issue. Omit if issue covers entire picture. | xs:decimal | 0..1 |

#### QCErrorSubtitle-type

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Element** | **Attribute** | **Definition** | **Value** | **Card.** |
|  |  |  |  |  |
| **QCErrorSubtitle-type** |  |  |  |  |
| StartTimecode |  | Track timeline where issue starts. Omit if issue exists for entire period or if start is unknown | manifest:Timecode-type | 0..1 |
| EndTimecode |  | Track timeline where issue ends. Omit, if problem persists to end of timeline or if end is unknown | manifest:Timecode-type | 0..1 |
| TimeOffset |  | For errors with alignment issues (e.g., subtitle Sync), the duration of offset. Negative means subtitle is ahead of video. | xs:duration | 0..1 |

#### QCErrorMetadata-type

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Element** | **Attribute** | **Definition** | **Value** | **Card.** |
| **QCErrorMetadata-type** |  |  |  |  |
| XPath |  | XPath reference to object with issue(s) | xs:anyURI | 0..1 |
| LineNumber |  | Line number in file of issue | xs:positiveInteger | 0..1 |

#### QCErrorArtwork-type

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Element** | **Attribute** | **Definition** | **Value** | **Card.** |
| **QCErrorArtwork-type** |  |  |  |  |
| XOffset |  |  |  | 0..1 |
| YOffset |  |  |  |  |
| XLength |  |  |  |  |
| YLength |  |  |  |  |

#### QCErrorPackage-type

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Element** | **Attribute** | **Definition** | **Value** | **Card.** |
| **QCErrorPackage-type** |  |  |  |  |
|  |  |  |  | 0..1 |

# Asset Status Manifest (ASM)

TBD

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Element** | **Attribute** | **Definition** | **Value** | **Card.** |
| **-type** |  |  |  |  |
|  |  |  |  |  |

# Asset Status

TBD

# Avail Confirmation

This belongs in Avails, not here. All types are there, not here. Also, if it evolves, it will evolve with Avails.

# NOTES

Special cases

* Indication that delivered content isn’t to spec (kind of a waiver).
* Need an indication of what is missing. For example, is forced dubs required for video.
* Ordering things
  + That exist (advertised to exist, or promised in a deal)
  + That don’t exist or might not exist, but are ‘standard’
  + Something special (e.g., special trailers or artwork)
* Capacity planning and delivery timing?
* Flows
  + Standard delivery flow
  + Exception flows