



entertainment merchants association

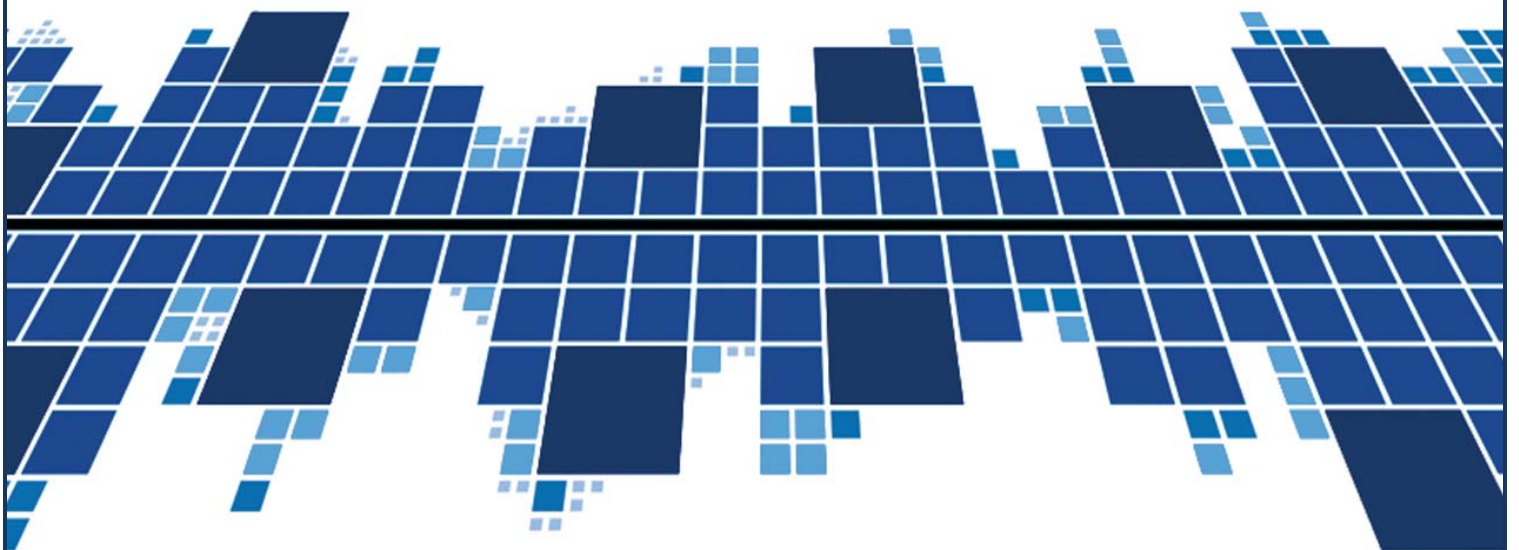
16530 Ventura Blvd., Suite 400

Encino, CA 91436

818.385.1500

[www.entmerch.org](http://www.entmerch.org)

**EMA Digital Supply Chain  
Standards, Specifications &  
Best Practices Suite  
for  
Digital Audio-Visual Distribution**





January 8, 2013

To All Stakeholders in the Digital Delivery of Audio Visual Entertainment –

The Entertainment Merchants Association (EMA) is proud to present this suite of standards, specifications and best practices as a toolkit to facilitate a more efficient and cost-effective digital supply chain.

Workflows in today's home entertainment digital supply chain have been manual and inconsistent. Much labor is being invested in the process; it is error-ridden; and new product is often slow getting to market.

Leading digital retailers have come together through EMA to identify those back-end functions that offer no competitive advantages or marketing differentiation. Five initial pain-points were identified and Work Groups were established:

- Metadata
- Mezzanine Files
- Image files
- Avails
- Closed Captioning

The standards, specifications, and best practices included in this toolkit were developed through extensive discussion, compromise, and collaboration. Participating retailers included Best Buy/CinemaNow, Google, Microsoft, Netflix, and Vudu.

EMA encourages each company to individually evaluate how to best implement this toolkit with your trading partners. The more broadly this toolkit is adopted by retailers, the more likely it is to be supported by content providers, increasing overall utility and efficiencies.

EMA's Digital Supply Chain Work Groups will continue to meet to review proposed changes and additions to this toolkit. EMA members should submit change/addition requests to Jennifer Lane Burnell at [jlane@entmerch.org](mailto:jlane@entmerch.org).

Let's use this suite of standards, specs and best practices to together build a more profitable digital distribution business.

All the best,

A handwritten signature in black ink, appearing to read 'Mark Fisher', with a long horizontal line extending to the right.

Mark Fisher  
Interim President



entertainment merchants association

16530 Ventura Blvd., Suite 400

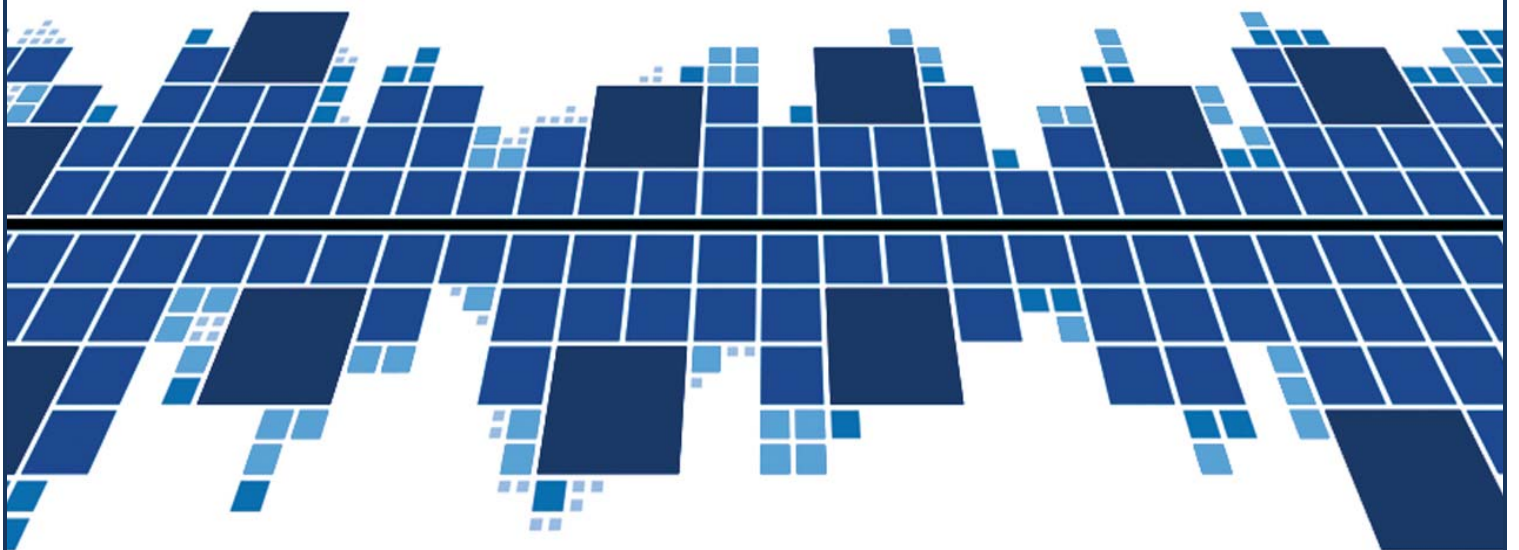
Encino, CA 91436

818.385.1500

[www.entmerch.org](http://www.entmerch.org)

# **Best Practices & Standards For the Delivery of Metadata for Digital Audio-Visual Distribution**

January 8, 2013







## INTRODUCTION

EMA, with the support of MovieLabs, initially created the EMA Metadata structure. This structure robustly described structure and data fields to communicate descriptive, logical, and technical metadata from content providers to retailers/distributors and ultimately to consumers. EMA Metadata, along with work being done by DECE in developing UltraViolet's metadata needs, is the basis for the all-inclusive "Common Metadata" as published by MovieLabs.

After further development, and more recent collaboration with the DEG Media and Content Operations Committee, the DEG-EMA Core v1.3 is herewith published. This Core describes those fields that are most critical to the retailer's ability to merchandise and distribute media assets. It is pared down from "Common Metadata" in order to ease marketplace adoption. However, individual adopters can elect to share additional metadata fields from "Common Metadata" as appropriate and as agreed.

### Why is it needed?

- Metadata related to digital video distribution is too often communicated manually between trading partners.
- It is also communicated in a variety of inconsistent structures and formats that are requested by retailers/distributors or as supplied by various content providers.
- Bad metadata that is translated to supplier catalogs or retailer/distributor websites can cause lost sales.
- Late metadata can bottleneck the supply chain, delaying product availability.

A governance structure, including EMA, DEG, UltraViolet, EIDR, and MovieLabs is being developed. EMA members are encouraged to contact Jennifer Lane at [jlane@entmerch.org](mailto:jlane@entmerch.org) with questions or suggested additions or changes.



## Media Entertainment Core Metadata

**'mdmec' namespace**

## CONTENTS

1	Introduction .....	1
1.1	Document Organization .....	1
1.2	Document Notation and Conventions .....	1
1.3	Normative References .....	1
1.4	Informative References.....	2
1.5	XML Namespaces .....	2
1.6	Identifiers .....	2
1.7	Status .....	2
2	MEC Core Metadata .....	3
2.1	CoreMetadata-type.....	3
2.1.1	Publisher-type .....	3
2.2	Common Metadata derived types.....	4
2.2.1	Basic Metadata Usage .....	5
2.2.2	Digital Asset Metadata Usage .....	7
2.2.3	Additional Usage Rules .....	8

**NOTE:** No effort is being made by EMA, the EMA Digital Council or Motion Picture Laboratories to in any way obligate any market participant to adhere to the Common Metadata or EMA Metadata. Whether to adopt the Common Metadata and/or EMA Metadata in whole or in part is left entirely to the individual discretion of individual market participants, using their own independent business judgment. Moreover, EMA, the EMA and Motion Picture Laboratories each disclaim any warranty or representation as to the suitability of the Common Metadata and/or EMA Metadata for any purpose, and any liability for any damages or other harm you may incur as a result of subscribing to this Metadata.



---

## REVISION HISTORY

Version	Date	Description
1.0	January 5, 2010	EMA Metadata
1.2	November 1, 2011	EMA Metadata with EMA Core definition
2.0	January 3, 2013	DEG-EMA Media Entertainment Core Metadata.

## 1 INTRODUCTION

The Entertainment Merchant's Association (EMA) and the Digital Entertainment Group (DEG) have defined metadata for the description of information delivered from Publishers to Retailers. This document was developed by the EMA Digital Council and the DEG Media and Content Operations Committee with the objective of standardizing the metadata communication from content providers to digital retailers.

This document defines Media Entertainment Core Metadata v2.0. This is also referred to as MEC Metadata, or MEC.

MEC Metadata builds upon EMA Metadata and Common Metadata developed by Motion Picture Laboratories (MovieLabs), EMA, DEG and others. Common Metadata includes elements that cover typical definitions of media, particularly movies and television. Common Metadata has two parts: Basic Metadata and Digital Asset Metadata. Basic Metadata includes descriptions such as title and artists. It describes information about the work independent of encoding. Digital Asset metadata describes information about individual encoded audio, video and subtitle streams, and other media included. Package and File Metadata describes a single possible packaging scenario and ties in other metadata types. Ratings and Parental Control information is described.

Common Metadata is designed to provide definitions to be inserted into other metadata systems, such as was done here, EIDR metadata and UltraViolet metadata. Selected elements of the Common Metadata are used in derived specifications. Adopters then define additional metadata to cover areas not included in Common Metadata.

### 1.1 Document Organization

This document is organized as follows:

1. Introduction—Background, scope and conventions
2. Core Metadata –Definition of MEC Metadata.

### 1.2 Document Notation and Conventions

The document uses the conventions of Common Metadata [CM].

### 1.3 Normative References

[CM] TR-META-CM MovieLabs Common Metadata, v2.0, <http://www.movielabs.com/md/md>

[EIDR-TO] *EIDR Technical Overview*, November 2010. <http://eidr.org/technology/#docs>

All Common Metadata references are included by reference.

---

## 1.4 Informative References

[DECEMD] DECE Content Metadata. <http://www.uvvu.com/techspec-archive.php>

## 1.5 XML Namespaces

This document defines:

- mdmec: includes Media Entertainment Core Metadata-specific data  
'mdmec' builds on
  - md: Common Metadata corresponding with Common Metadata [CM]

## 1.6 Identifiers

Identifiers must be universally unique. Recommended identifier schemes may be found in Common Metadata [CM] and in DECE Content Metadata [DECEMD].

The use of Entertainment Identifier Registry identifiers ([www.eidr.org](http://www.eidr.org)) is strongly encouraged. Please see [EIDR-TO].

## 1.7 Status

This specification is completed and ready for implementation. Although tested, we anticipate that additional implementation experience will yield recommendation for changes. Implementers should anticipate one or more revisions. Reasonable measures will be taken to ensure changes are backwards compatible. See Backwards Compatibility Best Practices in [CM]

## 2 MEC CORE METADATA

The section defines the MEC Metadata. The rules for what must be included and how it is encoded is in this section and its references.

Note that the structure accommodates additional data which may be included optionally.

### 2.1 CoreMetadata-type

This defines the MEC metadata, including both the descriptive information (Basic Metadata) and the encoding information (Physical metadata). It is as follows:

Element	Attribute	Definition	Value	Card.
CoreMetadata-type				
Basic		Basic Metadata	md:BasicMetadata-type	
DigitalAsset		Digital Asset Metadata: encoding information for the assets	md:DigitalAssetMetadata-type	1..n
TitleInternalAlias		Title used by involved parties to refer to this content	xs:string	0..1
Source		Organization that created the metadata	mdmec:Publisher-type	0..1
CompanyDisplayCredit		Organizations associated with the asset, for display purposes.	md:CompanyCredits-type	0..n
GroupingEntity		The "Network" or "Studio" that the product should be merchandised under within a retailer's website. For example, "Warner Bros".	md:GroupingEntity-type	0..n

#### 2.1.1 Publisher-type

Element	Attribute	Definition	Value	Card.
Publisher-type			md:OrgName-type (by extension)	
	organizationID	Organization Identifier for the publisher. This is an ID use by the Publisher to refer to itself.		0..1

	retailerSpecificID	Identifier by which the Retailer knows the Publisher	xs:string	0..1
DisplayName		Name of Publisher in a displayable form. This is the name intended to be presented to a consumer.	xs:string	
SortName		Name of Publisher intended for sorting purposes. It is not necessary to include SortName if it is identical to DisplayName.		0..1
ContactInfo		Contact information for the publisher	md:ContactInfo-type	

SortName is typically used when a Publisher has variations on its name that may not sort properly (e.g., some instances have a prefix).

## 2.2 Common Metadata derived types

Common Metadata [CM09] includes elements that cover typical definitions of media, particularly movies and television. Basic Metadata includes descriptions such as title and artists. It describes information about the work independent of encoding. Digital Asset metadata describes information about individual encoded audio, video and subtitle streams, and other media included. Package and File Metadata describes one possible packaging scenario and ties in other metadata types. Ratings and Parental Control information is described.

Common Metadata is designed to provide definitions to be inserted into other metadata systems, such as EMA's. Although EMA uses some element from Common Metadata, it also defines additional metadata to cover areas specific to EMA's requirements.

The following MEC types are derived directly from Common Metadata:

MEC Type	Common Metadata Type
mdmec:BasicMetadata-type	md:BasicMetadata-type
mdmec:DigitalAssetMetadata-type	md:DigitalAssetMetadata-type

All mandatory elements and attributes must be included. Any optional elements may be included. The following elements and attributes are required for MEC usage, regardless of whether they are optional, except as noted.

The following table uses the following conventions:

- Structure is given by table indentation. Parent level elements to the left.
- Attributes begin with '@'. For example, @ContentID refers to the ContentID attribute

## 2.2.1 Basic Metadata Usage

Element or Attribute		Usage Rules
BasicMetadata-type		Required
	@ContentID	Required
	UpdateNum	Shall be included if the record is an update (i.e., not the first record distributed)
	LocalizedInfo	At least one instance required
	@language	Required
	default	must be included for one instance of LocalizedInfo for the language of original production
	TitleDisplay60	Required. Note that TitleDisplay19 no longer required.
	TitleSort	Required
	OriginalTitle	Required
	Summary190	Description that is unique to that content
	Summary400	Recommended
	Cast	if applicable
	Genre	Exactly one primary genre shall be included. It will be from <a href="http://www.movelabs.com/md/mec/mec_primary_genre.html">http://www.movelabs.com/md/mec/mec_primary_genre.html</a> @source='http://www.movelabs.com/md/mec/mec_primary_genre.html'. @level='0'. Any additional genres may be included.
	ArtReference	At least one instance is mandatory, additional instances are optional
	CopyrightLine	Required
	AlternateTitle	Required
	RunLength	Specify to at least seconds. Zero is recommended for season and series.
	ReleaseYear	Required
	ReleaseDate	should include the highest date/time resolution available

ReleaseHistory ReleaseType	Original Release date must be included with ReleaseType='original'. When applicable Local Release date must be included with ReleaseType='local'. When applicable DVD Release date must be included with ReleaseType='DVD'
WorkType	Required
PictureColorType	optional, but it should be included
PictureFormat	optional, but it should be included
AltIdentifier	optional, but it should be included for all commonly used identifiers. For example, if ISAN is available, it should be included.
RatingSet	SHALL be included for all available ratings in the regions where Retailers are authorized to sell this content. All elements and attributes should be included if applicable to the rating. The condition attribute should be used if the primary purpose of the edit is a derivation from a parent for the purposes of ratings change (e.g., airline edit or 'unrated edition').
People	Include Actor(s), Director(s) and Producer(s) as applicable.
CountryOfOrigin	defined as the "generally accepted country of reference".
PrimarySpokenLanguage	Language should be included for the language(s) in which the video was shot (i.e., the language the "lips move to.") Movies such as Babel may have multiple PrimarySpokenLanguage elements. This should not be used for languages spoken incidentally and subtitled; for example, "RU" (Russian) in <i>The Hunt for Red October</i> .
SequenceInfo Parent	SHALL be included for the following work types: Season, Episode, Promotion, Excerpt, Supplemental
Number	Required
HouseID	Shall be used for production ID in episodic content
Parent	Shall be included for work type of Non-episodic Show if that show is part of a season or series. Should be included for derived works such as Director's Cut and promotional activity.

## 2.2.2 Digital Asset Metadata Usage

Element or Attribute		Usage Rules
DigitalAssetMetadata-type		Shall be included for each track included
	Audio	Required
	Type	Required
	Encoding	Required
	Codec	Required
	CodecType	The IANA namespace shall be used
	BitrateMax	Required
	SampleRate	Required
	SampleBitDepth	Required
	Language	Required
	Channels	Required
	Video	Required
	Type	Required
	Encoding	Required
	Codec	Required
	CodecType	The IANA namespace SHALL be used
	BitrateMax	Required
	Picture	Required
	AspectRatio	Required
	ColorType	Required
	SubtitleLanguage	Shall be included if the video contains visible subtitles.
	Subtitle	If applicable



		Format	Required
		Type	Required
		FormatType	Required
		Language	Required

### 2.2.3 Additional Usage Rules

- Original Release/Air Date (Year for features; Date for episodic television) – should be defined as the original release date in the target region of distribution.
- StartsWith search titles are included in TitleAlternate with type 'StartsWith'.
- ReleaseHistory should apply to distribution target.



This XML file does not appear to have any style information associated with it. The document tree is shown below.

---

```
<!--
  edited with XMLSpy v2013 sp1 (x64) (http://www.altova.com) by Craig Seidel
  (MovieLabs)
-->
<xs:schema xmlns:mdmec="http://www.movielabs.com/schema/mdmec/v2.0"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:md="http://www.movielabs.com/schema/md/v2.0/md"
  targetNamespace="http://www.movielabs.com/schema/mdmec/v2.0"
  elementFormDefault="qualified" attributeFormDefault="unqualified" version="1.0">
  <xs:import namespace="http://www.movielabs.com/schema/md/v2.0/md"
    schemaLocation="http://www.movielabs.com/schema/md/v2.0/md-v2.0.xsd"/>
  <!-- Media Entertainment Core (MEC) METADATA -->
  <!-- Verson 2.0 -->
  <xs:complexType name="BasicMetadata-type">
    <xs:complexContent>
      <xs:extension base="md:BasicMetadata-type"/>
    </xs:complexContent>
  </xs:complexType>
  <xs:complexType name="DigitalAssetMetadata-type">
    <xs:complexContent>
      <xs:extension base="md:DigitalAssetMetadata-type"/>
    </xs:complexContent>
  </xs:complexType>
  <xs:complexType name="Publisher-type">
    <xs:complexContent>
      <xs:extension base="md:OrgName-type">
        <xs:sequence>
          <xs:element name="ContactInfo" type="md:ContactInfo-type"/>
        </xs:sequence>
        <xs:attribute name="retailerSpecificID" type="xs:string"/>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
  <xs:complexType name="Disposition-type">
    <xs:sequence>
      <xs:element name="EntryType" type="xs:string"/>
      <xs:element name="EntryID" type="md:id-type" minOccurs="0"/>
      <xs:element name="IssueDate" type="md:YearDateOrTime-type" minOccurs="0"/>
      <xs:choice>
        <xs:element name="ReplacesEntryID" type="md:id-type" minOccurs="0"/>
        <xs:element name="ReplacesEntryDate" type="md:YearDateOrTime-type"
          minOccurs="0"/>
      </xs:choice>
      <xs:any namespace="##other" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="CoreMetadata-type">
    <xs:sequence>
      <xs:element name="Basic" type="mdmec:BasicMetadata-type"/>
      <xs:element name="DigitalAsset" type="mdmec:DigitalAssetMetadata-type"
        minOccurs="0" maxOccurs="unbounded"/>
      <xs:element name="TitleInternalAlias" type="xs:string" minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>
</xs:schema>
```

```
<xs:element name="TrackingID" type="xs:string" minOccurs="0"/>
<xs:element name="Source" type="mdmec:Publisher-type" minOccurs="0"/>
<xs:element name="CompanyDisplayCredit" type="md:CompanyCredits-type"
minOccurs="0" maxOccurs="unbounded"/>
<xs:element name="GroupingEntity" type="md:GroupingEntity-type" minOccurs="0"
maxOccurs="unbounded"/>
</xs:sequence>
</xs:complexType>
</xs:schema>
```

# Common Metadata 'md' namespace

---

## CONTENTS

1	Introduction .....	1
1.1	Overview of Common Metadata .....	1
1.2	Document Organization .....	1
1.3	Document Notation and Conventions .....	2
1.3.1	XML Conventions .....	2
1.3.2	General Notes .....	3
1.4	Normative References .....	4
1.5	Informative References.....	5
1.6	Best Practices for Maximum Compatibility.....	5
2	Identifiers .....	7
2.1	Identifier Structure .....	7
2.1.1	ID Simple Types .....	8
2.2	Asset Identifiers .....	8
2.2.1	ContentID .....	8
2.2.2	APID .....	10
2.3	Organization ID.....	11
3	General Types Encoding .....	12
3.1	Language Encoding.....	12
3.2	Region encoding.....	12
3.3	Date and Time encoding.....	12
3.3.1	Duration.....	13
3.3.2	Time .....	13
3.3.3	Dates and times .....	13
3.3.4	Date and time ranges .....	14
3.4	String encoding.....	14
3.5	Organization Naming and Credits.....	14
3.5.1	CompanyDisplayCredit-type.....	15
3.6	People Naming and Identification .....	15
3.6.1	PersonName-type.....	15
3.6.2	PersonIdentifier-type .....	16
3.7	Money-type and Currency .....	16
3.8	Role Encoding, Role-type .....	17
3.9	Keywords Encoding .....	17
3.9.1	Name/Value Pairs, NVPair-type, NVPairMoney-type .....	17
3.10	Personal/Corporate Contact Information, ContactInfo-type .....	18
3.11	Cryptographic Hash .....	18
3.12	GroupingEntity-type .....	19
3.13	Private Data.....	19
4	Basic Metadata .....	20
4.1	BasicMetadata-type .....	20
4.1.2	BasicMetadataInfo-type.....	26
4.1.3	ContentIdentifier-type .....	30

---

4.1.4	BasicMetadataPeople-type.....	30
4.2	Compilation Object .....	33
4.2.1	CompObj-type .....	34
4.2.2	CompObjID-type.....	34
4.2.3	CompObjData-type.....	34
4.2.4	Comp-ObjEntry-type.....	35
5	Digital Asset Metadata.....	37
5.1	Digital Asset Metadata Description.....	37
5.2	Definitions.....	37
5.2.1	DigitalAssetMetadata-type and DigitalAssetSet-type.....	37
5.2.2	DigitalAssetAudioData-type.....	38
5.2.3	DigitalAssetAudioEncoding-type .....	39
5.2.4	DigitalAssetVideoData-type.....	42
5.2.5	DigitalAssetVideoEncoding-type .....	43
5.2.6	DigitalAssetVideoPicture-type .....	46
5.2.7	DigitalAssetSubtitleData-type .....	48
5.2.8	DigitalAssetImageData-type .....	50
5.2.9	DigitalAssetInteractiveData-type .....	51
5.2.10	DigitalAssetWatermark-type .....	53
5.2.11	Cards.....	53
6	Container Metadata .....	56
6.1	Container Metadata Description .....	56
6.2	Definitions.....	56
6.2.1	ContainerMetadata-type .....	56
6.2.2	ContainerProfile-type.....	58
7	Content Ratings .....	59
7.1	Description.....	59
7.2	Rules .....	59
7.2.1	“Unrated” .....	59
7.3	Definition.....	59
7.3.1	ContentRating-type.....	59
7.3.2	ContentRatingDetail-type .....	61
8	Content Rating Encoding.....	62
9	Selected Examples .....	63
9.1	People Name Examples .....	63
9.2	Release History Example .....	67
9.3	Content Rating Examples .....	68
10	Redefine Support.....	70
10.1	General XML Type Redefines .....	70
10.2	Type-specific Redefines .....	70
10.2.1	Identifiers.....	70
10.2.2	Basic Metadata.....	71
10.2.3	Digital Asset Metadata.....	72
10.2.4	Content Ratings.....	74
10.2.5	Container Metadata .....	74

---



---

10.2.6	Compilation Object .....	74
10.2.7	Additional Types .....	75
10.2.8	Release History .....	75

**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





## Common Metadata

Ref: TR-META-CM  
Version: 2.0  
Date: January 3, 2013

---

to this Common Metadata.

---

## REVISION HISTORY

See *Common Metadata Revision History, TR-META-REV*, for detailed revision information.

Version	Date	Description
1.0	January 5, 2010	Original Version
1.1	January 6, 2011	Incorporates corrections.
1.2	November 1, 2011	Incorporates corrections and enhancements, primarily to support derived specifications.
1.2a-1.2e	May 29, 2012, September 24, 2012, October 11, 2012	Minor schema alignment (no schema changes), EIDR IDs, additions to controlled vocabularies, Ratings improvements, and minor corrections and additions.
1.2f	December 16, 2012	Moved Section 8 Content Ratings Encoding to a separate document: TR-META-CR, Common Metadata Content Ratings, <a href="http://www.movie labs.com/ratings">www.movie labs.com/ratings</a>
2.0	January 3, 2013	Major revision

---

## 1 INTRODUCTION

The B2B transfer of media requires metadata to describe that media. Several activities underway at the time of this document's authoring have metadata needs that overlap. This document in conjunction with associated XML schemas defines the content and one possible encoding of such data.

This is designed as a resource. Those using this specification may extend the definition with additional data element specific for their needs. They may replace elements with others perhaps more suitable to their needs; however, for interoperability all are highly encouraged to use the data elements exactly as defined.

### 1.1 Overview of Common Metadata

Common Metadata includes elements that cover typical definitions of media, particularly movies and television. Common Metadata has two parts: Basic Metadata and Digital Asset Metadata. Basic Metadata includes descriptions such as title and artists. It describes information about the work independent of encoding. Digital Asset metadata describes information about individual encoded audio, video and subtitle streams, and other media included. Package and File Metadata describes one possible packaging scenario and ties in other metadata types. Ratings and Parental Control information is described.

Common Metadata is designed to provide definitions to be inserted into other metadata systems. A given metadata scheme, for example, the Entertainment Merchant's Association (EMA) may select element of the Common Metadata to be used within its definitions. EMA would then define additional metadata to cover areas not included in Common Metadata.

### 1.2 Document Organization

This document is organized as follows:

1. Introduction—Provides background, scope and conventions
2. Identifiers—Specification of identifiers used to reference metadata.
3. General Types Encoding—Specific of encoding methods (e.g., language, region).
4. Basic Metadata—Content descriptive metadata definition
5. Digital Asset Metadata—Encoded media metadata definition
6. Container Metadata – Metadata describing media containers
7. Content Rating—Methods for encoding content ratings
8. Content Rating Encoding—Content Ratings can now be found in *Common Metadata Content Ratings* at [www.movie labs.com/ratings](http://www.movie labs.com/ratings).
9. Examples
10. Redefine Support – Information on using schema features to tightly control vocabulary

---

## 1.3 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.

### 1.3.1 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.

#### 1.3.1.1 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 `initialLowercaseAttribute`.
- XML structures are formatted as Courier New, such as `md:id-type`
- Names of both simple and complex types are followed with “-type”

### 1.3.1.2 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.

### 1.3.2 **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.

---

## 1.4 Normative References

- [TR-META-CR] *Common Metadata Content Ratings*, TR-META-CR, [www.movielabs.com/ratings](http://www.movielabs.com/ratings). Note that a specific version is not referenced as it is intended that the latest version will be used. Referencing specifications may select a specific version of the referenced document.
- [EIDR-TO] *EIDR Technical Overview*, November 2010. <http://eidr.org/technology/#docs>
- [RFC2141] R. Moats, *RFC 2141, URN Syntax*, May 1997, <http://www.ietf.org/rfc/rfc2141.txt>
- [RFC3629] Yergeau, F., et al, *RFC 3629, UTF-8, a transformation format of ISO 10646*, November, 2003. <http://www.ietf.org/rfc/rfc3629.txt>
- [RFC3986] Berners-Lee, T., et al, *RFC 3986, Uniform Resource Identifier (URI): Generic Syntax*, January 2005, <http://www.ietf.org/rfc/rfc3986.txt>
- [RFC5646] Philips, A, et al, *RFC 5646, Tags for Identifying Languages*, IETF, September, 2009. <http://www.ietf.org/rfc/rfc5646.txt>
- [IANA-LANG] IANA Language Subtag Registry. <http://www.iana.org/assignments/language-subtag-registry>
- [ISO3166-1] Codes for the representation of names of countries and their subdivisions -- Part 1: Country codes, 2007.
- [ISO3166-2] ISO 3166-2:2007 Codes for the representation of names of countries and their subdivisions -- Part 2: Country subdivision code
- [ISO4217] Currency shall be encoded using ISO 4217 Alphabetic Code. [http://www.iso.org/iso/currency\\_codes\\_list-1](http://www.iso.org/iso/currency_codes_list-1)
- [ISO8601] ISO 8601:2000 Second Edition, *Representation of dates and times, second edition*, 2000-12-15.
- [ISO26324] ISO26324:2012, *Information and documentation -- Digital object identifier system*.
- [47CFR9.103(c)(9)] “Closed caption decoder requirements for all apparatus.”, Title 47, part 71.103(c)(9) 2012, 47 CFR 9.103(c)(9), <http://ecfr.gpoaccess.gov/cgi/t/text/text-idx?c=ecfr&sid=53ad878c54cd79758c7fa602e4bc8975&rgn=div8&view=text&node=47:4.0.1.1.6.0.3.8&idno=47>. See also, Federal Register 77:62 (30 March 2012) p. 19480. <http://www.gpo.gov/fdsys/pkg/FR-2012-03-30/pdf/2012-7247.pdf>
- [SMPTE-428-3] SMPTE-428-3-2006, “D-Cinema Distribution Master Audio Channel Mapping and Channel Labeling”, 2006.
- [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/>

---

## 1.5 Informative References

[RFC4647] Philips, A., et al, *RFC 4647, Matching of Language Tags*, September 2006.  
<http://www.ietf.org/rfc/rfc4647.txt>

[EIDR] Entertainment Identifier Registry (EIDR), <http://eidr.org/resources/>  
European Broadcast Union, Tech 3295 – P\_META Metadata Library,  
[http://www.ebu.ch/en/technical/metadata/specifications/notes\\_on\\_tech3295.php](http://www.ebu.ch/en/technical/metadata/specifications/notes_on_tech3295.php)

The following metadata standards activities have numerous associated specifications. Rather than listing each specification, sites where specifications can be found are listed.

- AMPAS – Academy of Motion Picture Arts and Sciences  
<http://www.oscars.org/science-technology/council/projects/index.html>
- SMPTE Metadata Dictionary: <http://www.smp-te-ra.org/mdd/>
- MPEG – Motion Pictures Experts Group <http://mpeg.chiariglione.org/>
- MHP – DVB Multimedia Home Platform <http://www.mhp.org>
- CableLabs VOD Metadata 2.0 <http://www.cablelabs.com/specifications/md20.html>
- Dublin Core Metadata Initiative: <http://dublincore.org/>.
- TV Anytime (ETSI) <http://www.etsi.eu/WebSite/Technologies/TVAnytime.aspx>
- PBCore: [www.pbcore.org](http://www.pbcore.org)
- Vocabulary Mapping Framework: <http://cdlr.strath.ac.uk/VMF/index.htm>

## 1.6 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.



---

## 2 IDENTIFIERS

Identifiers and metadata are closely linked. In essence, all identifiers have corresponding metadata that describes the object being identified. Just as it is useful to distinguish between different kinds of objects with different kinds of identifiers, it is useful to distinguish the metadata in terms of those same objects.

The primary objects being identified and described in metadata are:

- Content – Content ID (ContentID)
- Encoded Stream – Physical Asset (Asset Physical ID; APID)

### 2.1 Identifier Structure

The primary requirement for identifiers is globally uniqueness. Individual systems using Common Metadata are free to use own identifiers as long as there is no identifier collision.

The following represents a structure for identifiers that should be used if specific usage does not specify otherwise. This structure is designed around the following principles

- Global uniqueness
- Coexistence of identifier schemes (ID Federation)
- Ability to use identifiers within a URL

Common Metadata identifiers use the general structure of the “urn:” URI scheme as discussed in RFC 3986 (URN) and RFC 3305 with a “md” namespace identifier (NID). However, for Common Metadata, rather than the fully articulated “urn:md” we abbreviate to “md:”. The basic structure for a Common Metadata ID is

`<MDID> ::= “md :”<type> “:”<scheme>“:”<SSID>`

- `<type>` is the type of identifier. These are defined in sections throughout the document defining specific identifiers.
- `<scheme>` is either a Common Metadata recognized naming scheme (e.g., “ISAN”) or “org” non-standard naming. These are specific to ID type and are therefore discussed in sections addressing IDs of each type.
- `<SSID>` (scheme specific ID) is a string that corresponds with IDs in scheme `<scheme>`. For example, if the scheme is “ISAN” then the `<SSID>` would be an ISAN number.

There is a special case where `<scheme>` is “org”. This means that the ID is assigned by a recognized organization within their own naming conventions. If `<scheme>` is “org” then

`<SSID> ::= <organization><UID>`

- `<organization>` is a unique name assigned to an organization, with the following rules:
  - Organization is defined as domain name. For example, movielabs.com becomes `md:org:movielabs.com:...`  and `bbc.co.uk`  becomes `md:org:bbc.co.uk:...`
  - Other naming schemes may be used in contexts where names can be assigned within the scope of ID usage.

- <UID> is a unique identifier assigned by the organization identified in <organization>. Organizations may use any naming convention as long as it complies with RFC 3986 syntax.

Some sample identifiers are

- ContentID: md:cid:EIDR:10.5240%2ffF592-58D1-A4D9-E968-5435-L
- Content ID: md:cid:ISAN: 0000-3BAB-9352-0000-G-0000-0000-Q
- Content ID: md:cid:org:MYSTUDIO:12345ABCDEF

### 2.1.1 ID Simple Types

The simple type md:id-type is the basic type for all IDs. It is XML type xs:anyURI. All identifiers are case insensitive and should be registered in canonical format and case sensitive identifiers should not be used.

The simple types ContentID-type AssetLogicalID-type and AssetPhysicalID-type are defined as md:id-type and can be used when a more specific designation is required.

## 2.2 Asset Identifiers

Content Identifiers are assigned by the content owner or its designee. The following scheme provides flexibility in naming while maintaining uniqueness.

Common Metadata defines two types of asset identifiers:

- A Content Identifier (ContentID) denotes an abstract representation of a content item.
- Asset Physical Identifier (APIID) refers to a physical entity (i.e., a file) that is associated with content.

### 2.2.1 ContentID

Syntax: "md:cid:"<scheme>":"<SSID>

A ContentID points to Basic metadata. ContentIDs may refer to abstract items such as shows or seasons, even if there is no separate asset for that entity. A ContentID must be globally unique.

The following restrictions apply to the <scheme> and <SSID> part of a ContentID:

- A ContentID scheme may not contain the colon character.
- Where display formats exists (i.e., human readable versus computer-readable) use display format.
- ContentID < scheme> and ContentID <SSID> shall be in accordance with Table 2-1. Additional schemes may be added in the future.

**Table 2-1: Content Identifier Scheme and Value**

Scheme	Expected value for <SSID>
ISAN	An <ISAN> element, as specified in ISO15706-2 Annex D.
TVG	TV Guide
AMG	AMG
IMDB	IMDB
MUZE	Muze
TRIB	Tribune
Baseline	Baseline Research ID, <a href="http://www.baselineresearch.com">www.baselineresearch.com</a>
UUID	A UUID in the form 8-4-4-4-12
URI	A URI; this allows compatibility with TVAnytime and MPEG-21
GRid	A Global Release identifier for a music video; exactly 18 alphanumeric characters
EIDR	Entertainment ID Registry. <a href="http://www.eidr.org">http://www.eidr.org</a> . In accordance with [ISO26324] and [EIDR-TO]
EIDR-S	Entertainment ID Registry. <a href="http://www.eidr.org">http://www.eidr.org</a> . EIDR-S is a shortened EIDR that does not include the "10.5240/" prefix.
EIDR-X	Entertainment ID Registry. <a href="http://www.eidr.org">http://www.eidr.org</a> . EIDR-X is an extended form of EIDR-S. EIDR-X is an EIDR-S form identifier followed by a colon (":") and an alphanumeric string.
ISRC	Master recordings, ISO 3901, <a href="http://www.ifpi.org/content/section_resources/isrc.html">http://www.ifpi.org/content/section_resources/isrc.html</a>
ISWC	Musical Works, <a href="http://www.cisac.org">http://www.cisac.org</a>
DOI	Digital Object Identifier <a href="http://www.doi.org">http://www.doi.org</a>
SMPTE-UMID	SMPTE-UMID as per SMPTE ST 330-2004
Ad-ID	Ad-ID as per format defined at <a href="http://www.ad-id.org/help/structure.cfm">http://www.ad-id.org/help/structure.cfm</a>
GTIN	Global Trade Item Number. <a href="http://www.gtin.info/">http://www.gtin.info/</a>
UPC	Universal Product Code (UPC). UPC-E should be converted to UPC-A form.

CRid	CRid (Content Reference Identifier) as per RFC 4078 <a href="http://tools.ietf.org/html/rfc4078">http://tools.ietf.org/html/rfc4078</a>
clDf	Content ID Forum. clDf Specification 2.0, Rev 1.1., 4/1/2007.
file	Indicates that the identifier that follows is a local file name.
org	<SSID> begins with the Organization ID of the assigning organization and follows with a string of characters that provides a unique identifier. The <ssid> must conform to RFC 3986 with respect to valid characters. In the absence of agreements between parties using IDs of this form, we recommend the use of an organization DNS domain (e.g., movielabs.com).

Identifiers that contain URI shall use Percent-Encoding as per [RFC3986] for characters not allowed in URNs as per [RFC2141]. For example, space (SP) is replaced by '%20' and slash ('/') is replaced by '%2f'. For example,

EIDR: 10.5240/F592-58D1-A4D9-E968-5435-L

ContentID: md:cid:EIDR:10.5240%2fF592-58D1-A4D9-E968-5435-L

## 2.2.2 APID

Syntax: "md:apid:< scheme>":"<SSID>[":"<extension>"]

An APID is constrained as follows:

- Each APID is globally unique

The following restrictions apply to the <scheme>, <SSID> and <extension> part of an APID:

- An APID scheme may not contain the colon character
- Where display formats exist (i.e., human readable versus computer-readable) use display format.
- APID < scheme> and APID <SSID> shall be structured the same as ContentID
- Optional <extension> is additional characters appended to the APID and may not contain colons

For example

- APID: md:apid:EIDR-S:58D1-A4D9-E968-F592-5435-M
- APID: md:apid:ISAN:0000-3BAB-9352-0000-G-0000-0000-Q:p1

Note that APIDs may be constructed from ContentIDs. For example:

- ContentID: `md:cid:org:MyCompany:ABCDEFGH`  
APID: `md:apid:org:MyCompany:ABCDEFGH:100`
- ContentID: `md:cid:ISAN:0000-3BAB-9352-0000-G-0000-0000-Q`  
APID: `md:apid:ISAN:0000-3BAB-9352-0000-G-0000-0000-Q:A203`

## 2.3 Organization ID

Common Metadata assumes one additional type be provided. That is an Organization ID (OrgID). `md:orgID-type` is a simple type of type `md:id-type`.

Currently, there is not an adequate global identification scheme, so this element should be used only if both the sending and receiving parties have an a priori agreement regarding the contents of this ID.

### 3 GENERAL TYPES ENCODING

#### 3.1 Language Encoding

Language shall be encoded in accordance with RFC 5646, *Tags for Identifying Languages* [RFC5646]. The subtags that are available for use with RFC 5646 are available from the Internet Assigned Numbers Authority (IANA) at [IANA-LANG] <http://www.iana.org/assignments/language-subtag-registry>.

Matching, if applicable, should be in accordance with RFC 4647, *Matching Language Tags*, [RFC4647].

The xs:language type shall be used for languages. Language should be as specific as possible; for example, ‘ja-kata’ is preferable to ‘ja’.

#### 3.2 Region encoding

Region coding shall use the ISO 3166-1 two-letter alpha-2 codes [ISO3166-1]. Informally described here: [http://en.wikipedia.org/wiki/ISO\\_3166-1\\_alpha-2](http://en.wikipedia.org/wiki/ISO_3166-1_alpha-2).

When subdivisions are required, ISO3166-2 shall be used [ISO3166-2]. Informally described here: [http://en.wikipedia.org/wiki/ISO\\_3166-2](http://en.wikipedia.org/wiki/ISO_3166-2).

Common Metadata shall use the following type for region:

Element	Attribute	Definition	Value	Card.
Region-type				
country		ISO 3166-1 Alpha 2 code	xs:string Pattern: "[A-Z][A-Z]"	(choice)
countryRegion		ISO 3166-2 Code	xs:string Pattern: "[A-Z][A-Z]-[0-9A-Z]+"	(choice)

The MadeforRegion-type simple type is a restriction of xs:string that allows country code, ‘Domestic’ or ‘International’. For example, it could be “US”, “Domestic” or “International”.

#### 3.3 Date and Time encoding

Date and time encoding shall use the XML rules. That is, where ISO 8601 [ISO8601] deviates from XML encoding, XML encoding shall apply.

### 3.3.1 Duration

Durations are represented using `xs:duration`. `xs:time` should not be used for duration.

### 3.3.2 Time

`xs:time` is used for a recurring time.

### 3.3.3 Dates and times

XML is fairly rigid in its date and time encoding rules. Specifically, it is difficult to have a single element where resolution may range from ‘year’ to ‘date’ to ‘time’. In some instances such as air dates/time, resolution might be year (movie released in 1939), date (movie released on December 25, 2009), or date and time (episode aired November 6, 2001, or November 6, 2001, 10:00 PM EST).

- Year encoding uses `xs:gYear` (Gregorian year)
- Date encoding (year, month and day) uses `xs:date`
- Date encoding that includes both date and time shall uses `xs:dateTime`

Time zone should be included with `xs:dateTime` elements to avoid ambiguity. If representing a single point in time with no relevant time zone, Coordinated Universal Time (UTC) should be used.

In some cases, there are options for including year, date and date-time. Optional elements should be included if known and relevant.

As of version 1.2 of this specification, a new type has been define to support elements that require year, date (year and day), or time (including date) without a priori knowledge of the resolution. This simple type is `YearDateOrTime-type`.

Element	Attribute	Definition	Value	Card.
<code>YearDateOrTime-type</code>		A simple type that syntactically allows the inclusion of a year, a date or a date-time.	<code>xs:union</code> with memberTypes of <code>xs:gYear</code> , <code>xs:date</code> , <code>xs:dateTime</code>	

### 3.3.4 Date and time ranges

Date Ranges may be encoded using the DateTimeRange-type:

Element	Attribute	Definition	Value	Card.
DateTimeRange				
Start		Start of time period	xs:dateTime	
End		End of time period	xs:dateTime	

### 3.4 String encoding

String lengths are specified in characters (rather than bytes) unless otherwise stated. A string using double-byte Unicode characters can result in string elements whose actual size in bytes is larger than the stated length.

### 3.5 Organization Naming and Credits

Organization names shall include both a user-friendly display name and a sortable name. If the display name and the sort name are the same, the SortName element may be excluded.

Element	Attribute	Definition	Value	Card.
OrgName-type				
	organizationID	Organization's unique ID	md:orgID-type	0..1
	idType	ID scheme used for organizationID	xs:string	0..1
DisplayName		General display format. Safest to use as it accommodates various permutation on the name.	xs:string	
SortName		Sortable version of name. This will often be last name first. This may be displayed.	xs:string	0..1
AlternateName		Other names for this organization	xs:string	0..n



## 3.5.1 CompanyDisplayCredit-type

This type describes the intended audience for metadata:

Element	Attribute	Definition	Value	Card.
MetadataCompanyCredits-type				
DisplayString		String to be displayed.	md:OrgName-type	0..n
	language	Language of DisplayString. If blank, then all languages	xs:language	0..1
Region		Region(s) for which credits apply.	md:Region-type	0..n
DisplaySequence		Order of display. Lower-numbered entries are displayed before higher-numbered entries. Entries without this element should be displayed after numbered entries.	xs:integer	0..1

## 3.6 People Naming and Identification

This section describes the internationalized naming approach used for encoding metadata. This section also defines person identification for the purposes of metadata.

### 3.6.1 PersonName-type

Element	Attribute	Definition	Value	Card.
PersonName-type				
DisplayName		Person's name for display purposes.	xs:string	1..n
	language	Language of DisplayName. There may be multiple instances of DisplayName, but only with unique language attributes.	xs:language	0..1
SortName		Name used to sort. May be excluded if identical to DisplayName.	xs:string	0..n
	language	Language of SortName. There may be multiple instances of SortName, but only with unique language attributes.	xs:language	0..1

FirstGivenName		First name	xs:string	0..1
SecondGivenName		Second name	xs:string	0..1
FamilyName		Family name	xs:string	0..1
Suffix		Suffix	xs:string	0..1
Moniker		Alternative name, usually of the form <FirstGivenName> "<Moniker>" <FamilyName> (e.g., <i>Scatman</i> in <i>Benjamin Sherman "Scatman" Crothers</i> ). Note, Moniker is misspelled but retained for backwards compatibility.	xs:string	0..1

### 3.6.2 PersonIdentifier-type

Assuming there is an identifier associated with the person, this structure holds information about that identifier.

Element	Attribute	Definition	Value	Card.
PersonIdentifier-type				
Identifier		Identifier associated with this individual within the Namespace	xs:string	
Namespace		Namespace for identifier.	xs:string	
ReferenceLocation		Location associated for the identifier within the namespace. This is expected to be an online reference to information about the individual.	xs:anyURI	

### 3.7 Money-type and Currency

Currency shall be encoded using ISO 4217 Alphabetic Code [ISO4217].

[http://www.iso.org/iso/currency\\_codes\\_list-1](http://www.iso.org/iso/currency_codes_list-1)

Element	Attribute	Definition	Value	Card.
Money-type				
	currency	Currency as expressed in ISO 4217 Currency Alphabetic Code. For example, 'USD' for US Dollars.	xs:string	
Value		Value	xs:decimal	

[ISO4217] typically allows two or three digits after the decimal. However, Value in this element may have as many decimal places as necessary.

### 3.8 Role Encoding, Role-type

Roles shall be encoded in accordance with 'Term' column of EBU Role codes found here: [http://www.ebu.ch/metadata/cs/web/ebu\\_RoleCodeCS\\_p.xml.htm](http://www.ebu.ch/metadata/cs/web/ebu_RoleCodeCS_p.xml.htm), plus "Other Group" and "Other" (referring to an unclassified individual).

Roles are defined in the simple type md:Role-type.

The JobFunction element allows for alternate schemes, however the scheme attribute is not supported at this time. At a future release, alternate schemes may be defined.

### 3.9 Keywords Encoding

Keywords are often culturally specific, so different keywords may exist for different regions. At this time, no keywords are defined.

#### 3.9.1 Name/Value Pairs, NVPair-type, NVPairMoney-type

Use of Name/Value pairs provides considerable flexibility for growth. The NVPair-type complex type allows for any additional business data to be included in tuple format.

Element	Attribute	Definition	Value	Card.
NVPair-type				
Name		Identification of the parameter being specified	xs:string	
Value		Value specified for Name.	xs:string	

NVPairMoney-type is like NVPair-type except the Value is currency-based.

Element	Attribute	Definition	Value	Card.
NVPairMoney-type				
Name		Identification of the parameter being specified	xs:string	
Value		Value specified for Name.	avail:Money-type	

### 3.10 Personal/Corporate Contact Information, ContactInfo-type

Element	Attribute	Definition	Value	Card.
ContactInfo-type				
Name		Person or point of contact	xs:string	
PrimaryEmail		Primary email address for user.	xs:string	
AlternateEmail		Alternate email addresses, if any	xs:string	0..n
Address		Mail address	xs:string	0..n
Phone		Phone number. Use international (i.e., +1 ...) format.	xs:string	0..n

### 3.11 Cryptographic Hash

The Hash-type definition describes a cryptographic hash such as SHA-1 and MD5.

Element	Attribute	Definition	Value	Card.
ContactInfo-type		Value of the cryptographic hash	xs:string	
	Method	The hash generation method.	xs:string	0..n

Values for method include:

- ‘MD2’, ‘MD4’, ‘MD5’ – Message Digest algorithms.
- ‘SHA-0’, ‘SHA-1’, ‘SHA-2’, ‘SHA-3’. SHA (Secure Hash Algorithm) family of algorithms. Distinction between hashes of different length is implicit in the hash and should not be mentioned specifically. For example, use ‘SHA-2’, not ‘SHA-224’.

### 3.12 GroupingEntity-type

Grouping Entity type allows logical grouping of assets. This is typically around studio or network, but it can be any logical content grouping.

Element	Attribute	Definition	Value	Card.
GroupingEntity-type				
Type		The type of the group.	xs:string	
GroupingIdentity		A string that uniquely identifies the group.	xs:string	
DisplayName		A string that will be displayed when referring to this group.	xs:string	1..n
	language	The language associated with the DisplayName. If language is absent, DisplayName applies to all langauges.	xs:language	0..1
Region		Region where group applies. If Region is absent, the group applies internationally.	md:Region-type	0..1

Type defines the type of grouping. Currently, the only defined value is “publisher”, although other values are not prohibited.

“publisher” indicates the grouping is around the organization publishing the content. Note that the actual publisher may differ from the publisher visible to the consumer. In that case, the GroupIdentity would reflect the actual publisher and the DisplayName would reflect the publisher familiar to the consumer.

### 3.13 Private Data

The following is defined to allow schemas using Common Metadata to extend elements with data specific to that use. Interoperability will be very limited, elements of this type should be used with extreme caution.

Element	Attribute	Definition	Value	Card.
PrivateData-type		Value of the cryptographic hash	xs:string	
(any)		Any data outside of ‘md’ namespace.	xs:any ##other	1..n

## 4 BASIC METADATA

Basic Metadata is a set of data that are essentially ubiquitous in content systems. They may be used throughout.

### 4.1 BasicMetadata-type

Element	Attribute	Definition	Value	Card.
BasicMetadata-type				
	ContentID	Content ID in Section 2.	md:ContentID-type	
UpdateNum		Version. Initial release should be 1. This is a value assigned by the metadata creator that should only be incremented if a new version of metadata is released. If absent, 1 is to be assumed. This is assigned by the metadata originator.	xs:int	0..1
LocalizedInfo		Instances of localized metadata.	md:BasicMetadataInfo-type	1..n
RunLength		Approximate Runlength of the referenced work (not the original product). Resolution SHALL be at least minutes. Resolution should be seconds or better. For a season or series, this should either be zero or the typical length of an episode. For broadcast, this should be the content length (e.g., an hour show with commercials might have a 44 minute RunLength).	xs:duration	
ReleaseYear		The year of original release. This applies to the version that is being released.	xs:gYear	
ReleaseDate		Year, Date or Date and Time of release or original air date. Adds month and day information to ReleaseYear. The year part of ReleaseDate must match ReleaseYear.	md:YearDateOrTime	0..1
ReleaseHistory		Information about releases	md:ReleaseHistory-type	0..n

Element	Attribute	Definition	Value	Card.
WorkType		Type of the work. See Work Type Enumeration.	xs:string	
WorkTypeDetail		More specific definition of Work Type to allow a more detailed description	xs:string	0..1
PictureColorType		Color type of asset. This SHALL not be included for audio-only assets.	md:ColorType-type	0..1
PictureFormat		A textual description of the aspect ratio format type, as defined below. This field does not contain the actual aspect ratio.	xs:string	0..1
ThreeD		Indicates whether work is in 3D. 'true' means 3D, 'false' or absent means not 3D.	xs:boolean	0..1
AspectRatio		Aspect ratio of active pixels, the form m:n (e.g., 4:3,16:9, 2:35:1)	xs:string	0..1
AltIdentifier		Other identifiers for the same content.	md:ContentIdentifier-type	0..n
RatingSet		All ratings associated with this content	md:ContentRating-type	0..1
People		People involved in production, with the exception of alternate language-specific roles (e.g., voice talent for language dubbing)	md:BasicMetadataPeople-type	0.. n
CountryOfOrigin		The country from where the title originates, ISO3166-1 e.g., "US" for United States. A derived would should refer to the country of the original work.	md:Region-type	0..1
PrimarySpokenLanguage		Primary spoken language of original production. As guidance this can be considered, "The language lips move to."	xs:language	0..n
OriginalLanguage		The original language of the production.	xs:language	0..n
VersionLanguage		The language, if any, associated with this particular version.	xs:language	0..n

Element	Attribute	Definition	Value	Card.
AssociatedOrg		Organization associated with the asset in terms of production, distribution, broadcast or in another capacity (see below for roles).	md:OrgName-type	0..n
	role	Role of the associated organization.	xs:string	0..1
SequenceInfo		Indicates how asset fits into sequence	md:ContentSequenceInfo-type	0..1
Parent		Metadata for parent items. Note that this is recursive.	Md:BasicMetadataParent-type	0..n

#### 4.1.1.1 WorkType and WorkTypeDetail Enumerations

WorkType shall be enumerated to one of the following (categories are to support the definition, but are not included in the enumeration). WorkTypeDetail is stated where applicable.

Music related:

- ‘Album’ – A collection of songs
- ‘Song’
- ‘Music Video’ – Music Video, not ‘Performance’
- ‘Ring Tone’
- ‘Other Music’

Film related:

- ‘Movie’ – A full length movie regardless of distribution (e.g., theatrical, TV, direct to disc, etc.) and content (e.g., includes documentaries).
- ‘Short’ – a film of length shorter than would be considered a feature film.

TV, web and mobile related:

- ‘Series’ – a show that might span one or more seasons or might be a miniseries.
- ‘Season’ – a season of a Series. It will contain one more episodes.
- ‘Episode’ – an episodes of a season or miniseries. A pilot is also an episode. If episode is a ‘webisode’, ‘mobisode’ or other specialized sequence, it should be noted in Keywords.
- ‘Non-episodic Show’ – TV or other show that is non-episodic; for example, sports and news.
  - Acceptable WorkTypeDetail includes ‘Sports’ and ‘News’.



- ‘Promotion’ – promotional material associated with media. This includes teasers, trailers, electronic press kits and other materials. Promotion is a special case of ‘Ad’.
  - Acceptable WorkTypeDetail includes ‘Trailer’, ‘Teaser’ and ‘EPK’.
- ‘Ad’ – any form of advertisement including TV commercials, informercials, public service announcements and promotions not covered by ‘Promotion’. This does not include movie trailers and teasers even though they might be aired as a TV commercial.

Other:

- ‘Excerpt’ – An asset that consists primarily of portion or portions of another work or works; for example, something having the ‘isclipof’ or ‘iscompositeof’ relationship.
- ‘Supplemental’ – Material designed to supplement another work. For example, and extra associated with a Movie for a DVD.
- ‘Collection’ – A collection of assets not falling into another category. For example, a collection of movies.
- ‘Franchise’ – A collection or combination of other types, for example, a franchise might include multiple TV shows, or TV shows and movies.

Although there is some overlap with Genre, Work Type is not language or culturally specific. Although terms may overlap, the usage does not. For example, the Work Type of ‘Sport’ refers to the capture of a sporting event, where a documentary on sport would have the ‘Non-episodic Show’ work type.

#### 4.1.1.2 ColorType-type

md:ColorType-type enumerates the picture color types. The enumerations are as follows:

- ‘color’ for color. If the work contains color, but is not clearly classified into one of the other categories, it should use the ‘color’ type.
- ‘bandw’ for black and white
- ‘colorized’ for colorized video (i.e., different from the original that is typically black and white).
- ‘composite’ for color composite (e.g., “Sin City”).
- ‘unknown’ for assets based on legacy metadata where color type is not specified.

#### 4.1.1.3 Picture Format Encoding

PictureFormat may be one of the following:

- ‘Letterbox’ – horizontal bars or other background appear above and/or below the picture’s active pixels.

- 
- ‘Pillarbox’ – vertical bars or other background appear to the left and/or right of the picture’s active pixels.
  - ‘Full’ – The active pixels fit the full area of the picture (within a few pixels). The entire original image is substantially included. This should not be confused with *fullscreen*, a term that may also refer to Pan and Scan.
  - ‘Stretch’ – The active pixels fit the full area of the picture (within a few pixels). The entire original image is substantially included. The image has been visibly stretched in one dimension to fit (e.g., a 4:3 image stretched to 16:9 frame).
  - ‘Pan and Scan’ – The active pixels the fit full area of the picture (within a few pixels). Part of the original image is not included. This includes fixed cropping, pan-and-scan and other cropping methods.
  - ‘Other’ – A picture format encoding other than the above applies. For example, ‘Smilebox’.

#### 4.1.1.4 UpdateNum

UpdateNum is an integer rather than a string (e.g., “2.3.1”) to simplify ordering. The Content Provider SHALL issue updates with increasing numbers.

#### 4.1.1.5 OriginalLanguage, PrimarySpokenLanguage, and VersionLanguage

OriginalLanguage is the language associated with the original production of the work. OriginalLanguage usually corresponds with PrimarySpokenLanguage, although not necessarily. A silent movie with title cards would have OriginalLanguage associated with those title cards. Anime from Japan would be considered ‘jp’, even though other language audio tracks might be available. There may be multiple OriginalLanguage elements if more than one language is associated with work. For example, the movie *Babel* has multiple OriginalLanguage elements. The movie *Hunt for Red October* would have one OriginalLanguage (i.e., English), even though there is spoken Russian. Mel Brooks *Silent Movie* would have an OriginalLanguage of English.

PrimarySpokenLanguage is a Primary spoken language spoken in the original production. That is, the language spoken by the actors, or more specifically, the language in which their lips are moving. It should include usage for meaningful dialog, but not an occasional word. For example, the movie *Babel* has multiple PrimarySpokenLanguage elements. The movie *Hunt for Red October*, would have two: English and Russian. The movie *Silent Movie*, even with one word spoken, would have no PrimarySpokenLanguage elements.

VersionLanguage refers to the particular version of the work. This should only exist if the title was edited for a particular language release.

#### 4.1.1.6 AssociatedOrg

The AssociatedOrg element provides information about organizational entities involved in the production, distribution, broadcast or other function relating to the asset. Often organizations provide different functions, so multiple organizations can be listed. The role attribute to AssociatedOrg may have one of the following values:

- ‘producer’ – involved in the production of the asset
- ‘broadcaster’ – network associated with asset’s broadcast
- ‘distributor’ – entity involved with distribution
- ‘editor’ - editor
- ‘encoding’ – entity that encodes media
- ‘post-production’ – entity that performs post-production functions, not in another category
- ‘other’ – any organization that does not fall into the previous categories.

#### 4.1.1.7 Release Information Encoding, ReleaseHistory-type

ReleaseType may include the following values:

- ‘original’ – first worldwide
- ‘Broadcast’
- ‘DVD’
- ‘Blu-ray’
- ‘Hospitality’
- ‘PayTV’ – Premium TV
- ‘InternetBuy’ – Offered for purchase on the Internet.
- ‘InternetRent’ – Offered for rent on the Internet.
- ‘Theatrical’
- ‘VOD’ – Home VOD.

This list may be expanded.

Element	Attribute	Definition	Value	Card.
ReleaseHistory-type				
ReleaseType		Release type as described above	xs:string	
	wide	Whether this release is a wide release, particularly for theatrical	xs:boolean	0..1
DistrTerritory		Where it was released to	md:Region-type	0..1

Date		When title was released. This may be a year, a date or a date and time. Generally, date is preferred over year. Date-time is preferred for broadcast air dates.	md:YearDateOrTime-type	
	scheduled	Date is assumed to be an actual date unless scheduled is included and holds the value 'true'	xs:boolean	0..1
Description		Description of the release,	xs:string	0..1
ReleaseOrg		Organization involved with this release.	md:OrgName-type	0..n

## 4.1.2 BasicMetadataInfo-type

This contains language-specific descriptive information.

In accordance of RFC5646, language may be inclusive of both language and character set. If submission uses more than one language or more than one character set, then multiple instances of this element may need to be supplied.

Element	Attribute	Definition	Value	Card.
BasicMetadataInfo-type				
	language	Language for this set of metadata as defined in Section 3.1. language should be as specific as possible (e.g., 'ja-kata' instead of 'ja').	xs:language	
	default	Indicates whether this is a language to use if no other available language is meaningful within the usage context (e.g., the native language for the user). 'true' indicates yes. 'false' or absence indicates no.	xs:boolean	0..1
TitleDisplay19		A brief version of the feature title (for display) that is up to a maximum length of 19 chars. All UIs SHOULD be able to support display of this field.	xs:string	0..1
TitleDisplay60		An alternate display version from TitleBrief for those UIs that can support longer fields than 19 Characters. This title may be up to 60 characters.	xs:string	0..1
TitleDisplayUnlimited		A display title with no length limit. It is recommended this be limited to no more than 256 characters.	xs:string	0..1

TitleSort		A sortable version of the feature title, e.g., "Incredibles, The" separated by commas.	xs:string	
ArtReference		Reference to art image	xs:anyURI	0..n
	resolution	String in the form <i>colxrow</i> (e.g., 800x600 would mean an image 800 pixels wide and 600 pixels tall).	xs:string	
Summary190		The title description – sentence. (max 190 char)	xs:string	
	cast	Flag to indicate if cast is or is not included in summary description. Missing assumes 'false'.	xs:boolean	0..1
Summary400		The title description -one paragraph, could be used as description in EPG. (max 400 char)	xs:string	0..1
	cast	Flag to indicate if cast is or is not included in summary description. Missing assumes 'false'.	xs:boolean	0..1
Summary4000		The title description – multi-paragraph. (max 4000 char)	xs:string	0..1
	cast	Flag to indicate if cast is or is not included in summary description. Missing assumes 'false'.	xs:boolean	0..1
Display Indicators		Indicators that MAY affect UI display. See Display Indictor Encoding below.	xs:string	0..n
Genre		Subject-matter classification of the show. See Genre Encoding below.	xs:string	0..n
	source	Naming system from which genre is derived.	xs:anyURI	0..1
	id	Identifier for genre used within source	xs:string	0..1
	level	Indicates precedence of genre, with a lower number being high precedence.	xs:integer	0..1
Keyword		Keyword	xs:string	0..n
VersionNotes		A descriptive statement about the reason why this cut was created or what its content represents with reference to other versions of this work. Do not include information about the language of the title in this field. If the cut is for a censor in a particular linguistic region, the region associated with the censor or censor name should be used, i.e., German censor version. VersionNotes may include edits for content, such as "Airplane Version".	xs:string	0..1

Region		The ISO 3166-1 code used to represent the name of the region(s) where the work is intended to be broadcast or shown. The code should be interpreted in a case insensitive manner. Note: Do not use the code "ww" to represent a worldwide region.	md:Region-type	0..1
OriginalTitle		Original title (no size limits).	xs:string	0..1
CopyrightLine		Displayable copyright line.	xs:string	0..1
PeopleLocal		People involved in the localized production, typically local voice actors.	md:BasicMetadataPeople-type	0..n
TitleAlternate		Alternate titles	xs:string	0..n
	type	Type of alternate title	xs:string	0..1
	language	The language of TitleAlternate if different from language attribute for BasicMetadataInfo-type.	xs:language	0..1

#### 4.1.2.1 Display Indicator Encoding

The values used for Display Indicator are at the discretion of the Publisher and the Retailer. Examples of values conceived for this element include, “CC”, “DVS”, “P” (season premiere) and “F” (finale).

#### 4.1.2.2 Genre Encoding

Genre is culturally and contextually specific, so different genre classifications may exist for different regions. This section presents a few alternatives for genre enumeration. Others will apply. Any genre list may be used.

The source attribute should be used for defined genre sets. It should be a URL that uniquely and unambiguously identifies a genre classification system. Ad hoc genre naming systems should use a URI whose namespace is under the control of the author.

The ‘id’ attribute may be used when a genre has an ID associated with the text string. In this case, the text string goes in the element, and the ID goes in the attribute. For example, using the EBU metadata, when the genre is “Surfing” the ID is “3.2.6.10”.

The level attribute indicates which genres are primary genre, main genre, subgenre, etc. A lower number indicates a higher precedence. Primary should be encoded as ‘0’. Only one instance of a primary should be included for a source.

Following are some genre encoding that may be used. Others may be used as well.

Alternatively, the following genres apply:

Region (Language)	Source	'source' attribute value
United States, Canada (English)	Common Metadata	<a href="http://www.movielabs.com/md/md/common_genre.html">http://www.movielabs.com/md/md/common_genre.html</a>
United States, Canada (English)	Library of Congress, Motion Picture and Television Reading Room	<a href="http://www.loc.gov/rr/mopic/miggen.html">http://www.loc.gov/rr/mopic/miggen.html</a>
Europe	European Broadcast Union (EBU) Tech 3295 – P_META Metadata Library, v 2.0, EBUContentGenre	<a href="http://www.ebu.ch/metadata/cs/web/ebu_ContentGenreCS_p.xml.htm">http://www.ebu.ch/metadata/cs/web/ebu_ContentGenreCS_p.xml.htm</a>
United States, Canada (English), TV	Cable Television Laboratories (CableLabs)	<a href="http://www.cablelabs.com/projects/metadata/downloads/genre_classification_list.pdf">http://www.cablelabs.com/projects/metadata/downloads/genre_classification_list.pdf</a>
Worldwide	Media Entertainment Core (MEC) from Entertainment Merchant's Association (EMA) and Digital Entertainment Group (DEG)	<a href="http://www.movielabs.com/md/mec/mec_primary_genre.html">http://www.movielabs.com/md/mec/mec_primary_genre.html</a>

#### 4.1.2.3 TitleAlternate/type Encoding

The following types should be used for the type attribute describing an AlternateTitle element:

- 'alternative' – legitimate alternative titles used to refer to the work. Shortened titles and common variations are included in this category. Most alternate titles are in the category.
- 'misspelling' – the title with a common misspelling. This should be included rarely, and a comprehensive list of possible misspellings should *not* be included.
- 'StartsWith' – search-centric title using convention typically used in Japan

For the movie “Terminator 2: Judgment Day”, ‘alternate’ titles could include “T2”, “Terminator 2”, “Terminator Two”, and “Judgment Day”. A ‘misspelling’ title could include “Terminator 2: Judgement Day”.

Alternate titles may include a language attribute to indicate a language different from the language attribute in the parent element. For example, if the LocalizedInfo is in Hiragana and there is a Katakana TitleAlternate with a type of “StartsWith”, the language should be ‘ja-kata’.

### 4.1.3 ContentIdentifier-type

This is designed to provide a cross reference to all other identifiers associated with this content. ContentIdentifier-type is a simple type based on md:id-type.

Namespace will be any namespace as listed in Table 2-1.

Element	Attribute	Definition	Value	Card.
ContentIdentifier-type				
Namespace		Namespace of identifier from Content ID table in the Identifiers section.	xs:string	
Identifier		Value of identifier.	xs:string	
Location		Reference location for item in the referenced namespace.	xs:anyURI	0..1

### 4.1.4 BasicMetadataPeople-type

Element	Attribute	Definition	Value	Card.
BasicMetadataPeople-type				
Job		Description of job function and, if applicable, character(s)	md:BasicMetadataJob-type	1..n
Name		Person or entity's name	md:PersonName-type	
Identifier		Formal identifier for this individual.	md:PersonIdentifier-type	0..n
Gender		Female, Male, Neutral, plural (name for group)	xs:string: "male", "female", "neutral" "plural"	0..1

#### 4.1.4.1 BasicMetadataJob-type

Element	Attribute	Definition	Value	Card.
BasicMetadataJob-type				
JobFunction		Role in production of media. Role is encoded in accordance with "Role Encoding" above. This version is displayable, but JobDisplay is preferred if present.	md:Role-type	



	scheme	The Role Scheme if alternate role systems are used.	xs:string	0..1
JobDisplay		Displayable version of Role. This allows metadata encoder to be more specific. For example, while JobFunction allows encoding of "Assistant Cameraman", JobDisplay could be "1 <sup>st</sup> Assistant Cameraman".	xs:string	0..n
	language	Language of JobDisplay. There may be multiple instances of JobDisplay, but only with unique language attributes.	xs:language	0..1
BillingBlockOrder		Order of listing, starting with 1. If missing, implies infinity and may be listed in any order. This need not be contiguous.	xs:int, [1..maxint]	0..1
Character		For actors, what role(s) they are playing. May be more than one.	xs:string	0..n
Guest		Whether this is a guest role (e.g., guest actor). If 'true', Job is as a guest. 'false' or absent is not guest.	xs:boolean	0..1

#### 4.1.4.2 BasicMetadataParent-type

This allows parent metadata to be included either by inclusion or reference. Usage rules will define if and when ParentContentID may be used in lieu of Parent. This is an optimization to avoid repeating full metadata sets when multiple objects have the same parent.

Element	Attribute	Definition	Value	Card.
BasicMetadataParent-type				
	relationshipType	The relationship between this asset and its parent as defined below.	xs:string	0..1
Parent		The parent metadata object.	md:BasicMetadata-type	(choice)
ParentContentID		Same as Parent, although included by reference instead of inclusion.	md:ContentID-type	(choice)

The relationshipType attribute may have the following enumerations:

- 'isclipof' – The asset is a subset of the larger body that is a contiguous subset of the parent. It may include unique small amounts of pre- and post-material such as new titles and credits. A typical example is a clip extracted from a larger video.
- 'isepisodeof' – The asset is an instance of an ordered sequence (i.e., an episode)

- “isseasonof” – The asset is a season and the parent is a show
- ‘ispartof’ – The asset is one complete segment of a larger body not covered by other definitions here. This may include a movie that is part of a series of movies. A song will be part of an album.
- ‘isderivedfrom’—The asset is a modification of the parent work. Some examples include a colorized version derived from a B&W version, and an edit such as a “Director’s Cut” or “Unrated Edition”.
- ‘iscompositeof’ – Asset includes a subset of the parent, such as may be found in a mashup. This contrasts a clip which is a proper subset otherwise unmodified.
- ‘issupplementto’ – is supplemental material. For example, outtakes and makings-of would be supplements.
- ‘ispromotionfor’ – is promotional material, such as a trailer. This is used when the child object has a work type of ‘Promotion’ and it is a promotion for the parent object.

Note that the parent object supplements information in the work—there is no structural or implied inheritance. When parents exist, they should be included either directly or by reference in each BasicMetadata-type element instance.

#### 4.1.4.3 ContentSequenceInfo-type

Describes Sequence, if part of sequence (episode, season, etc.). The actual sequence type is defined by the WorkType element.

Either Number or HouseSequence must be included. An element with HouseSequence but no number indicates the asset is non-sequenced and the HouseSequence is included for reference. This might be the case for a documentary whose airing sequence is irrelevant but the HouseSequence is still usable for management of the asset.

If neither Number nor HouseSequence is included, the ContentSequenceInfo-type based element should not be included.

Element	Attribute	Definition	Value	Card.
ContentSequenceInfo-type				
Number		Where it fits in sequence (e.g., episode 1 is “1”). Start with 1. If it is the only one in the sequence, it is numbered 1. Generally, sorting is done by Number. This is a strict ordering that may not necessarily correspond with the actual release number. For example, if a show issues with episodes represented in DistributionNumber as ‘1’, ‘2’, ‘3a’, ‘3b’ and 4, the corresponding Number will be ‘1’, ‘2’, ‘3’, ‘4’ and ‘5’.	xs:int	

DistributionNumber		A flexible, but mainly numeric, representation of the sequence of release within a set or season as used in distribution. Note that ContentNumber-type allows non-numeric values such as '3a' and '1.2'.	md:ContentNumber-type	0..1
HouseSequence		Identifier used internally for the asset. This might not be ordered the same as Number and is general in format allowing the inclusion of season or other information, e.g. 'S03E15' or 'GT0315' This is sometimes called Production ID.	xs:string	0..1
AlternateNumber		Another identifier by which this item is known, e.g. a number used by a distributor, such as a network, that does not fall into the above definitions. It also is general in format and may include season or other information.	md:ContentNumber-type	0..n

#### 4.1.4.4 ContentNumber-type

Element	Attribute	Definition	Value	Card.
ContentNumber-type		Content number contained as a string. Although this will typically be a number, it can also take other forms such as '3a' or '1.2'. This field should be suitable for display, but is assumed not to be sortable.	xs:string	
	domain	The namespace domain for the element.	xs:string	0..1

## 4.2 Compilation Object

A Compilation Object is a grouping outside of the structure of Basic Metadata (i.e., Parent definitions). Compilation Objects may include metadata, either by inclusion or reference. The md:CompObj-type is designed as a simple list of entries. It is intended for inclusion within other structures. The md:CompObjData-type is a more standalone structure that has an ID and a DisplayName field at the top level, and then the entries. Lists of entries are ordered. For example, if the entries are season premieres of a given show, they can be ordered in season order; and that ordering should be preserved.

#### 4.2.1 CompObj-type

The following defines a compilation:

Element	Attribute	Definition	Value	Card.
CompObj-type				
Entry		An individual entry in the compound object. The list is ordered.	md:CompObjEntry-type	1..n
CompilationClass		A description of the compilation	xs:string	0..1
	hasOtherInclusions	Indicates whether Entry elements include entries beyond the scope of the CompilationClass. Only applies if 'true'.	xs:boolean	0..1

#### 4.2.2 CompObjID-type

This is a simple type of type md:id-type that can be used to assign a unique identifier.

#### 4.2.3 CompObjData-type

Element	Attribute	Definition	Value	Card.
CompObjData-type			md:CompObj-type	(extension)
	CompObjID	Identifier for this compound object	md:CompObjID-type	0..1
DisplayName		A description of the Compound Object. There may be one entry per language.		0..n
	language	Language of the DisplayName in accordance with encoding described in Section 3.1.	xs:language	0..1

## 4.2.4 Comp-ObjEntry-type

Element	Attribute	Definition	Value	Card.
CompObjEntry-type				
DisplayName		A description of the Compilation Object. There may be one entry per language.		0..n
	language	Language of the DisplayName in accordance with encoding described in Section 3.1.	xs:language	0..1
EntryNumber		Represents the sequence of this entry relative to other entries. When specified, EntryNumber reflects order. This shall be numeric unless the system using this element specifically allows other formats (e.g., EIDR allows forms such as '1a').	xs:string	0..1
EntryClass		Describes the relationship of this Entry to the elements of the compilation.	xs:string	0..1
Entry		An individual entry in the compound object. The list is ordered.	md:CompObjEntry-type	0..n
ContentID		Content ID for item in the Compilation Object. It is assumed the metadata associated with this ContentID is available, and this field is used as an optimization to avoid repeating metadata.	md:ContentID-type	(choice)
BasicMetadata		Basic Metadata for the entry.	md:BasicMetadata-type	(choice)
(any)		Provisions for external references or other metadata (reserved).	(any##other)	(choice)

Metadata is included either by inclusion (use of BasicMetadata element) or by reference (use of ContentID element). Use of ContentID is an optimization for situations where the metadata for that ContentID is already provided. External systems, such as EIDR, can have external references. Therefore, additional elements can be used in lieu of ContentID or BasicMetadata. Within Common Metadata usage, only ContentID and BasicMetadata is used.

### 4.2.4.1 EntryClass Encoding

EntryClass defines how an element relates to the compilation. If the entity does not fit one of the following, this element should be omitted. Vocabulary is:

- ‘Episode’ – the item is an episode, or treated as an episode in the context of this compilation
- ‘Installment’ – the item is part of a sequential but non-episodic set of items
- ‘Part’ – the item is a piece of a large work, e.g. Part 1 and Part 2 of a film
- ‘Season’ – the item is a season of a series, or treated as a season in the context of this compilation

## 5 DIGITAL ASSET METADATA

Digital Asset Metadata describes includes relating to the Physical Asset that is distinct from the Logical Asset.

### 5.1 Digital Asset Metadata Description

A Digital Asset has certain properties that are not general to the Logical Asset and are therefore distinct from Basic Metadata. Digital Asset Metadata describes these properties. These data are distinct from Basic Metadata. The set of Digital Asset Metadata does not attempt to include all possible data about the Asset, only a subset of those most useful.

Metadata includes:

- Audio/video Encoding information
- Resolution, codec, frame rate, max bitrate

### 5.2 Definitions

#### 5.2.1 DigitalAssetMetadata-type and DigitalAssetSet-type

Element	Attribute	Definition	Value	Card.
DigitalAssetMetadata-type				
Audio		Metadata for an audio asset	md:DigitalAssetAudioData-type	(choice)
Video		Metadata for a video asset	md:DigitalAssetVideoData-type	(choice)
Subtitle		Metadata for subtitles	md:DigitalAssetSubtitleData-type	(choice)
Image		Metadata for Images	md:DigitalAssetImageData-type	(choice)
Interactive		Metadata for Interactive	md:DigitalAssetInteractiveData-type	(choice)

Element	Attribute	Definition	Value	Card.
DigitalAssetSet-type				
Audio		Metadata for an audio asset	md:DigitalAssetAudioData-type	0..n
Video		Metadata for a video asset	md:DigitalAssetVideoData-type	0..n

Subtitle		Metadata for subtitles	md:DigitalAssetSubtitleData-type	0..n
Image		Metadata for Images	md:DigitalAssetImageData-type	0..n
Interactive		Metadata for Interactive	md:DigitalAssetInteractiveData-type	0..n

## 5.2.2 DigitalAssetAudioData-type

Element	Attribute	Definition	Value	Card.
DigitalAssetAudioData-type				
Description		Description of the track. Description should be in the language given by the "Language" element below.	xs:string	0..1
Type		The type of track. See Audio Track Encoding. If not present, track is assumed to be 'primary'.	xs:string	0..1
Language		Language for the audio track as defined in Section 3.1.	xs:language	
	dubbed	If present and true, indicates Language is dubbed audio.	xs:boolean	0..1
Encoding		Audio encoding information. If CODEC is not known, this should not be included.	md:DigitalAssetEncoding-type	0..1
Channels		Number of audio channels, either as an integer (e.g., 2) or of the form x.y where x is full channels, and y is limited channels (e.g. "5.1")	xs:string	0..1
TrackReference		Track cross-reference to be used in conjunction with container-specific metadata.	xs:string	0..1
TrackIdentifier		Identifiers, such as EIDR, for this track. Multiple identifiers may be included.	md:ContentIdentifier-type	0..n
Private		Extensibility mechanism to accommodate data that is private to given usage.	md:PrivateData-type	0..1



### 5.2.2.1 Type Encoding

If Type is present, it should have one of the following values:

- ‘primary’ – primary audio track. There may be multiple primary tracks, with one for each language
- ‘narration’ - The visually impairment associated service is a complete program mix containing music, effects, dialogue, and additionally a narrative description of the picture content. The narration service may be coded using multiple channels. A Descriptive Video Service® (DVS®) track is a narration track.
- ‘dialogcentric’ - The hearing impaired associated service is a complete program mix containing music, effects, and dialogue with dynamic range compression. The dialogcentric service may be coded using multiple channels.
- ‘commentary’ – Commentary on the video. May be paired with a PIP.
- ‘other’ – not one of the above

### 5.2.3 DigitalAssetAudioEncoding-type

Element	Attribute	Definition	Value	Card.
DigitalAssetAudioEncoding-type				
Codec		Name of supported codec. See Codec encoding below.	xs:string	
CodecType		Formal reference identification of CODEC. See below	xs:string	0..n
BitrateMax		Peak Bitrate (bits/second) averaged over a short period.	xs:integer	0..1
BitrateAverage		Bitrate averaged over the entire track.	xs:integer	0..1
VBR		Variable BitRate information.	xs:string	0..1
SampleRate		Sample Rate (samples/second)	xs:integer	0..1
SampleBitDepth		Number of bits per audio sample	xs:integer	0..1
ChannelMapping		Indication of how channels are mapped to intended speaker locations.	xs:string	

Watermark		Information about watermark(s) embedded in audio.	md:DigitalAssetWatermark-type	0..n
(any)		Any additional elements	any##other	0..n

### 5.2.3.1 Audio CODEC Encoding

The following values should be used for elementary stream CODECs listed. “Other” should be used if the CODEC is not on the list. This list may be expanded over time.

- ‘AAC’ – Advanced audio CODEC
- ‘AAC-LC’
- ‘AAC-LC+MPS’
- ‘AAC-SLS’
- ‘AC-3’ – Dolby Digital, AC-3
- ‘AIFF’ – Audio Interchange File Format (when specific CODEC is not known)
- ‘ALAC’ – Apple Lossless Audio Codec
- ‘AMR’ – Adaptive MultiRate
- ‘DOLBY-TRUEHD’
- ‘DSD’ – Direct Stream Digital
- ‘DST’ – Direct Stream Transfer
- ‘DTS’ – DTS CODEC
- ‘DTS-ES’ – DTS ES (Extended Surround)
- ‘DTS-EXPRESS’ – DTS Express Audio
- ‘DTS-HRA’ – DTS-HD High Resolution Audio
- ‘DTS-96/24’ – DTS 96/24
- ‘DTS-MA’ – DTS-HD Master Audio
- ‘E-AC-3’ – Enhanced AC3, Dolby Digital Plus (DD+)
- ‘FLAC’ – Free Lossless Audio Codec
- ‘HE-AACv2’ – High Efficiency AAC v2
- ‘LPAC’ – Lossless Predictive Audio Compression
- ‘LTAC’ – Lossless Transform Audio Compression
- ‘MP3’ – MPEG 1 Layer 3
- ‘MPEG1’ – MPEG1 Layer 2
- ‘MPEG-4-ALS’
- ‘MLP’ – Meridian Lossless Package
- ‘PCM’ – Pulse Code Modulation, or Linear PCM

- ‘QCELP’ - Qualcomm Code Excited Linear Prediction
- ‘RealAudio-Lossless’ – Real Networks’ lossless format
- ‘Vorbis’ – Ogg Vorbis
- ‘WAV’ – used when specific CODEC (e.g., PCM) is unknown or not listed
- ‘WMA’ – Windows Media Audio
- ‘WM9-lossless’

### 5.2.3.2 CodecType Encoding

CodecType allows a more formal encoding of CODEC type based on formal registries. CodecType takes the form

<namespace> + ‘:’ + <codec type>

<namespace> is accordance with the following table:

Namespace	Definition	Reference for <codec type>
mpeg4ra	MPEG 4 Registration Authority	<a href="http://www.mp4ra.org/codecs.htm">http://www.mp4ra.org/codecs.htm</a>
IANA	Internet Assigned Numbers Authority (IANA) Audio Media Types	<a href="http://www.iana.org/assignments/media-types/audio/">http://www.iana.org/assignments/media-types/audio/</a>
rfc4281	CODEC encoded in according with RFC4281	<a href="http://www.ietf.org/rfc/rfc4281.txt">http://www.ietf.org/rfc/rfc4281.txt</a>

Only one entry per namespace is allowable.

### 5.2.3.3 VBR Encoding

The following values should be used for VBR:

- ‘VBR’ – Quality-based, 1-pass VBR
- ‘Constrained VBR’ – Constrained VBR, with maximum bitrate reflected in BitrateMax.
- ‘2-pass VBR’ – 2-pass, unconstrained VBR

### 5.2.3.4 ChannelMapping Encoding

The following values should be used for ChannelMapping when describing a single track. Their meaning is defined in [SMPTE-428-3]:

- ‘Left’, ‘Center’, ‘Right’, ‘LFE screen’, ‘Left surround’, ‘Right surround’, ‘Center surround’, ‘Left center’, ‘Right center’, ‘LFE 2’, ‘Vertical height front’, ‘Top center surround’, ‘Left wide’, ‘Right wide’, ‘Rear surround left’, ‘Rear surround right’, ‘Left surround direct’, ‘Right surround direct’.

Also applicable are applicable when ChannelMapping describes multiple tracks

- ‘stereo’ – Left and Right
- ‘5.1 Matrix’ – 5.1 channels matrixed in two channels
- ‘surround’ – Greater than two channels, without a specific channel assignment
- ‘L,R,C,LFE,LS,RS’
- ‘L,C,R,LS,RS,LFE’

## 5.2.4 DigitalAssetVideoData-type

Element	Attribute	Definition	Value	Card.
DigitalAssetVideoData-type				
Description		Description of this video track	xs:string	0..1
Type		Type of video track. If Type is missing, ‘primary’ is assumed. See Video Track Type encoding below.	xs:string	0..1
Encoding		Details on Video Encoding. If CODEC is unknown, this element should not be included.	md:DigitalAssetVideoEncoding-type	0..1
Picture		Picture description	md:DigitalAssetVideoPicture-type	
ColorType		Color type of video. Note that Color Type is also included in BasicMetadata, however, this provides information down to the individual stream.	md:ColorType-type	0..1
PictureFormat		PictureFormat of video. If absent, assumed to be same as in BasicMetadata. Note that PictureFormat is also included in BasicMetadata, however, this provides information down to the individual stream.	xs:string	0..1
SubtitleLanguage		Indicates the presence of subtitles embedded in the video stream, either closed (e.g., EIA-608B) or rendered into the video. This is distinguished from subtitles handled via separate tracks. Subtitles in separate tracks should be included in DigitalAssetMetadata-type’s Subtitle element. Language encoding is defined in Section 3.1. Silent	xs:language	0..n

		movies with text displays are considered 'normal' subtitles.		
	closed	Indicates whether captions are closed.	xs:boolean	0..1
	type	Indicates type of subtitle. See Section 5.2.7.1.	xs:string	0..1
SignedLanguage		Indicates the presence of signed language in the video. Language must be a sign language such as 'ase' for American Sign Language or 'fsl' for French Sign Language.	xs:language	0..1
CardsetList		Cardsets, such as distribution logos and anti-piracy notices, embedded in video.	md:DigitalAssetCards etList-type	0..n
TrackReference		Track cross-reference to be used in conjunction with container-specific metadata.	xs:string	0..1
TrackIdentifier		Identifiers, such as EIDR, for this track. Multiple identifiers may be included.	md:ContentIdentifier-type	0..n
Private		Extensibility mechanism to accommodate data that is private to given usage.	md:PrivateData-type	0..1

### 5.2.4.1 Video Type Encoding

Type, if present, should have one of the following values:

- 'primary' – primary video track. Whether or not this has burned-in subtitled is determined by the presence of the SubtitleLanguage element
- 'overlay' – PIP or other overlay track, intended for use with a 'primary' track
- 'angle' – alternate angle track
- 'other' - not one of the above

### 5.2.5 DigitalAssetVideoEncoding-type

Element	Attribute	Definition	Value	Card.
DigitalAssetVideoEncoding-type				
Codec		CODEC used. See Video CODEC Encoding below.	xs:string	
CodecType		Formal reference identification of	xs:string	0..n

		CODEC. See below		
MPEGProfile		MPEG Profile	xs:string	0..1
MPEGLLevel		MPEG Level (e.g., "3", "4", "1.3")	xs:string	0..1
BitrateMax		Bitrate (bits/second)	xs:integer	0..1
Watermark		Information about watermark(s) embedded in video.	md:DigitalAssetWatermark-type	0..n
(any)		Any additional elements	any##other	0..n

### 5.2.5.1 Video CODEC Encoding

The following values should be used for elementary stream CODECs listed. 'Other' should be used if the CODEC is not on the list. This list may be expanded over time.

- 'AVI Uncompressed'
- 'CineForm HD'
- 'DIVX'
- 'DV' – DV, including variants such as DVCPRO, DVCAM, etc.
- 'H.264' – H.264, MPEG-4 Part 10
- 'JPEG2000' – JPEG 2000, ISO/IEC 15444
- 'MOBICLIP' – Actimagine's Mobiclip CODEC
- 'MPEG1' – MPEG 1 Part 2
- 'MPEG2' – MPEG 2 Part 2
- 'On2' – On2 CODEC when not VP6, VP7 or VP8, or exact CODEC is unknown.
- 'PHOTOJPEG'
- 'PRORES' – Apple ProRes
- 'PRORESHQ' – Apple ProRes HQ
- 'PRORES422' – Apple ProRes 422
- 'QT Uncompressed' – Apple QuickTime Uncompressed
- 'REAL' – RealNetworks' RealVideo
- 'Spark' – Sorenson Spark
- 'SVQ' – Sorenson Video Quantizer

- ‘WMV’ – Windows Media Video when not WMV7, WVM8 or WMV9 or exact CODEC is unknown
- ‘WMV7’ – Windows Media Video 7
- ‘WMV8’ - Windows Media Video 8
- ‘WMV9’ – Windows Media Video 9
- ‘VC1’ – Microsoft VC-1
- ‘VP6’ – On2 VP6
- ‘VP7’ – On2 VP7
- ‘VP8’ – On2 VP8
- ‘XVID’ – Xvid
- ‘OTHER’ – not one of the above.

#### 5.2.5.2 CodecType Encoding

CodecType allows a more formal encoding of CODEC type based on formal registries. CodecType takes the form

<namespace> + ':' + <codec type>

<namespace> is accordance with the following table:

Namespace	Definition	Reference for <codec type>
mpeg4ra	MPEG 4 Registration Authority	<a href="http://www.mp4ra.org/codecs.htm">http://www.mp4ra.org/codecs.htm</a>
IANA	Internet Assigned Numbers Authority (IANA) Audio Media Types	<a href="http://www.iana.org/assignments/media-types/audio/">http://www.iana.org/assignments/media-types/audio/</a>

Only one entry per namespace is allowable.

### 5.2.6 DigitalAssetVideoPicture-type

Element	Attribute	Definition	Value	Card.
DigitalAssetVideoPicture-type				
AspectRatio		Aspect ratio of picture after decode (i.e., frame size). Note that this is not necessarily the original aspect ratio. These will be of the form n:m, for example, "16:9". The following should be used for the respective standard encoding: "16:9" "4:3", "1.85:1", "2.35:1", "1:1", etc.	xs:string	0..1
PixelAspect		Class of pixel aspect ratios	xs:string "square" "NTSC": "PAL" "other"	0..1
WidthPixels		Number of columns of pixels encoded (e.g., 1920)	xs:int	0..1
HeightPixels		Number of rows of pixels encoded (e.g., 1080)	xs:int	0..1
ActiveWidthPixels		Number of active pixels. Must be less than or equal to HeightPixels.	xs:int	0..1
ActiveHeightPixels		Number of active pixels. Must be less than or equal to WidthPixels.	xs:int	0..1
FrameRate		Frames/second. If interlaced, use the frame rate (e.g., NTSC is 30).	xs:int	0..1
	multiplier	This attribute indicates whether the 1000/1001 multiple should be applied. There is only one legal value for this attribute which is "1000/1001". If present, then apply 1000/1001 multiplier to FrameRate. For example, a FrameRate of 30 with multiplier='1000/1001' defines an actual frame rate of 29.97. If the frame rate is integral, this attribute shall not be present	xs:string "1000/1001"	0..1
	timecode	Indication of how drop frames are handled in timecode. See below.	xs:string	
Progressive		Whether image is progressive. "true"=progressive, "false"=interlaced	xs:boolean	0..1
	scanOrder	Indicates the scan order.	xs:string	0..1



ColorSubsampling		Color subsampling model, if applicable.	xs:string	0..1
Colorimetry		Picture colorimetry.	xs:string	0..1
Type3D		Type of 3D picture. Encoding currently undefined.	xs:string	0..1
(any)		Any additional elements.	any##other	0..n

### 5.2.6.1 FrameRate/timecode Encoding

The timecode element of FrameRate is encoded with the following values

- ‘Drop’ – Drop frame SMPTE timecode is used.
- EBU – AES/EBU embedded timecode
- Other – Other timecode

### 5.2.6.2 scanOrder Encoding

For interlaced (i.e., Progressive=‘false’), valid values are

- ‘TFF’ for Top Field First
- ‘BFF’ for Bottom Field First

For Progressive (i.e., Progressive=‘true’), it is recommend the optional attribute not be included. If included, the only valid is

- ‘PPF’ Picture Per Field

### 5.2.6.3 ColorSubsampling Encoding

Valid values for ColorSubsampling are:

- ‘4:1:1’
- ‘4:2:0’
- ‘4:2:2’
- ‘4:4:4’

### 5.2.6.4 Colorimetry Encoding

Values for Colorimetry include:

- ‘601’ – ITU Recommendation BT.601, *Studio encoding parameters of digital television for standard 4:3 and wide screen 16:9 aspect ratios* <http://www.itu.int/rec/R-REC-BT.601/en>
- ‘709’ – ITU Recommendation BT.709, *Parameter values for the HDTV standards for production and international programme exchange.* <http://www.itu.int/rec/R-REC-BT.709/en>

- ‘2020’ – ITU Recommendation BT.2020, *Parameter values for ultra-high definition television systems for production and international programme exchange*. <http://www.itu.int/rec/R-REC-BT.2020/en>
- ‘P3’ – SMPTE PR 431-2:2011 D-Cinema Quality – Reference Projector and Environment. This is also referred to as DCI-P3 or P3.

## 5.2.7 DigitalAssetSubtitleData-type

Element	Attribute	Definition	Value	Card.
DigitalAssetSubtitleData-type				
Format		Format of subtitle. See Subtitle Format Encoding below.	xs:string	0..1
	SDImage	Are subtitle images targeted towards SD included? ‘true’ means yes, ‘false’ or absent means no. This only applies if Format is ‘Image’ or ‘Combined’	xs:boolean	0..1
	HDImage	Are subtitle images targeted towards HD included? ‘true’ means yes, ‘false’ or absent means no. This only applies if Format is ‘Image’ or ‘Combined’	xs:boolean	0..1
Description		Description of this subtitle track. Description is in the language of the Language element.	xs:string	0..1
Type		Intended purpose or purposes of subtitle	xs:string	1..n
FormatType		Identification of subtitle format. See below	xs:string	0..1
Language		Language. See Language Encoding in Section 3.1.	xs:language	
Encoding		Encoding information (to be defined).	xs:anyType	0..1
CardsetList		Cards, such as distribution logos and anti-piracy notices, included in subtitle.	md:DigitalAssetCardsetList-type	0..n
TrackReference		Track cross-reference to be used in conjunction with container-specific metadata.	xs:string	0..1
TrackIdentifier		Identifiers, such as EIDR, for this track. Multiple identifiers may be included.	md:ContentIdentifier-type	0..n
Private		Extensibility mechanism to accommodate data that is private to given usage.	md:PrivateData-type	0..1

---

#### 5.2.7.1 Subtitle Type Encoding

Type describes the intended use or uses of the subtitle. If the track has more than one intended use, then multiple instances of Type must be included. For example, a single track might be used for both ‘normal’ and ‘SDH’ uses.

The following values may be used:

- ‘normal’ – subtitle used for languages
- ‘SDH’ – Subtitles for deaf and hard-of-hearing.
- ‘large’ – subtitles for visually impaired
- ‘forced’ – used to indicate subtitles is required regardless of whether the user has enabled subtitles. The correct language subtitle track must be chosen. Often referred to as ‘forced captions.’ A Type of ‘forced’ must only be used in conjunction with other Type instances, when the track contains a mix of forced and non-forced subtitles. Generally, when ‘forced’ is used it is the only instance of Type.
- ‘commentary’ – commentary, such as associated with a commentary audio track.
- ‘easyreader’ – ‘easy reader’ subtitle complying with US Federal requirements [47CFR9.103(c)(9)] . The ‘easy reader’ and ‘SDH’ Types are independent. That is, if a track is both ‘easy reader’ and ‘SDH’ it should be tagged with both Types.
- ‘other’ – subtitles for commentary, or other purposes.

#### 5.2.7.2 Subtitle Format Encoding

It is anticipated that IANA or others will provide a registry for subtitle encoding schemes. At that time, this section will be revised to reflect a more standard means of describing the subtitle. In the meantime, the following values may be used for Subtitle /Format:

- ‘Text’ – text subtitle
- ‘Image’ – image/picture data
- ‘Combined’ – Subtitle encoding that includes both text and image

#### 5.2.7.3 FormatType Encoding

FormatType may be one of the following:

- ‘3GPP’ – 3GPP Timed Text, MPEG 4 Part 17 Timed Text, ISO/IEC 14496-17.
- ‘Blu-Ray’
- ‘DCI’ – DCI Subtitle, SMPTE 428-7-2007 D-Cinema Distribution Master - Subtitle

- ‘DVB’ – DVB Subtitling, *ETSI 300 743 ‘Digital Video Broadcasting (DVB); Subtitling systems (2006-11)*
- ‘DVD’
- ‘SMPTE 21052-1 Timed Text’ – Timed Text Format (SMPTE-TT), SMPTE ST 2052-1:2010
- ‘SRT’ – SRT Subtitles
- ‘TTML’ – W3C Timed Text Markup Language (TTML) 1.0, W3C Recommendation 18 November 2010. <http://www.w3.org/TR/ttaf1-dfxp/>
- ‘WebVTT’ – WebVTT (Web Video Text Tracks)

## 5.2.8 DigitalAssetImageData-type

Element	Attribute	Definition	Value	Card.
DigitalAssetImageData-type				
Width		Number of columns of pixels (e.g., 1920)	xs:int	
Height		Number of rows of pixels (e.g., 1080)	xs:int	
Encoding		MIME type indicating encoding method	xs:string	
Language		Language(s) for this image, if any.	xs:language	0..n
TrackReference		Track cross-reference to be used in conjunction with container-specific metadata.	xs:string	0..1
TrackIdentifier		Identifiers, such as EIDR, for this track. Multiple identifiers may be included.	md:ContentIdentifier-type	0..n
Private		Extensibility mechanism to accommodate data that is private to given usage.	md:PrivateData-type	0..1

### 5.2.9 DigitalAssetInteractiveData-type

Note that this area is somewhat experimental and will likely change in the future. Please communicate any use cases that are not accommodated by the following.

Element	Attribute	Definition	Value	Card.
DigitalAssetInteractiveData-type				
Type		Type of interactive track (TBD).	xs:string	
FormatType		The form the encoding takes: text, executable or metadata.	xs:string	0..1
Language		Language. See Language Encoding in Section 3.1.	xs:language	0..1
Encoding		Encoding information.	md:DigitalAssetInteractiveEncoding-type	0..n
TrackReference		Track cross-reference to be used in conjunction with container-specific metadata.	xs:string	0..1
TrackIdentifier		Identifiers, such as EIDR, for this track. Multiple identifiers may be included.	md:ContentIdentifier-type	0..n
Private		Extensibility mechanism to accommodate data that is private to given usage.	md:PrivateData-type	0..1

For schema redefine support, the first three elements are defined in md:DigitalAssetInteractiveBaseData-type which is in turn defined as md:DigitalAssetInteractiveBaseData-group. This has no XML impact.

#### 5.2.9.1 Interactive Type Encoding

Type allows the following values:

- ‘Menu’ – Menu system for navigating settings, value added material and other options.
- ‘Standalone Game’ – Playable game that runs independently of audio or video material
- ‘Overlay Game’ – Game synchronized to audio or video material
- ‘Skins’ – Information that customizes appearance
- ‘Interactivity’ – Ability to choose settings, value added material and other options outside of menus. For example, pop-ups.
- ‘Other’

## 5.2.9.2 Interactive FormatType Encoding

FormatType allows the following values:

- ‘Text’ – Instructive text.
- ‘Executable’ – Software that is executable through a runtime environment. See Interactive RuntimeEnvironment.
- ‘Metadata’ – Declarative data that describes behavior to a runtime environment

## 5.2.9.3 Interactive Encoding Type

Element	Attribute	Definition	Value	Card.
DigitalAssetInteractiveEncoding-type				
RuntimeEnvironment		The execution runtime environment for the interactive content.		
FirstVersion		Earliest version of RuntimeEnvironment in which this encoding will play. If it plays in all versions, or all versions less than or equal to LastVersion, this element may be omitted.	xs:string	0..1
LastVersion		Last version of RuntimeEnvironment in which this encoding will play. If it plays in all versions, or all versions after FirstVersion, this element may be omitted	xs:string	0..1
(any)		Any other addition element(s)	xs:any##other	0..n

RuntimeEnvironment must use the following values when the associated runtime environments are used for ‘Executable’ and ‘Metadata’ FormatType values.

The following are runtime environments for Executable and Metadata Format Types.

- ‘CMX’ – Connected Media Experience
- ‘Flash’ – Adobe Flash
- ‘BD-J’ – Blu-ray Java
- ‘MHEG’ – MHEG-5, or more formally ISO/IEC 13522-5.
- ‘HTML5’ – W3C HTML5
- ‘Other’ – may be used when there is not a type convention.

### 5.2.10 DigitalAssetWatermark-type

Identification watermarks contain information that identifies content. This complex type describes which watermark is used and also includes information used for recognition.

Element	Attribute	Definition	Value	Card.
IDWatermark-type				
Vendor		Organization associated with watermark.	xs:string	
ProductAndVersionID		Identification of specific watermark version of the technology. It must be sufficiently precise to differentiate between incompatible watermarks from the same Vendor.	xs:string	
Data		Data is a string that either contains the information encoded by the watermark or is a reference to that data. Its content is outside the scope of this document. This may be vendor-private data.	xs:string	0..1

The combination of Vendor and ProductAndVersionID unambiguously identifies a watermark technology.

Vendor is a representation of a watermark vendor or relevant technology. This is not a strict enumeration to allow new vendors to be added. However, it is important that vendor names are used consistently. As general guidance, use initial caps (except for acronyms) and no spaces or punctuation. Following are a few examples:

- ‘Philips’
- ‘Civolution’
- ‘Verance’
- ‘Nielsen’
- ‘AACs’

ProductAndVersionID identifies the precise version of the technology. In particular, it is used to differentiate between incompatible watermarks from the same Vendor.

### 5.2.11 Cards

Cardsets are collections geographic-region-specific static text or graphics visually displayed. Typical cardsets are anti-piracy warnings. Cardsets may be embedded in video (i.e., burned in) or overlaid on video via a subtitle. Video-embedded cards are described as part of Video metadata. Subtitle cards are described as part of Subtitle metadata.

DigitalAssetCardsetList-type is used to describe a collection of cardsets used together. DigitalAssetCard-type are related by purpose.

Note that we use the term cardsets to refer to one or more cards. For example, a US anti-piracy cardset might consist of more than one warning card including an FBI card, a Department of Homeland Security card and others. Note also that a single video or subtitle track may contain multiple cardsets.

Note that this area is somewhat experimental and will likely change in the future. Please communicate any use cases that are not accommodated by the following.

### 5.2.11.1 DigitalAssetCardsetList-type

Element	Attribute	Definition	Value	Card.
Cardset-type				
Type		The intended general usage of the cardset list.	xs:string	0..n
Region		Location for which cardset is intended. For example, US anti-piracy cardset would be for country='us'. CardsetLists may apply to more than one region. If absent, cards are assumed to be worldwide.	md:MadeForRegion-type	0..n
CardSet		Description of the cardset.	md:DigitalAssetCardset-type	1..n

Type is encoded as follows:

- ‘Theatrical’ – Theatrical
- ‘Broadcast’ – Broadcast, not including Internet
- ‘Hospitality’ – Hospitality, such as airline and hotel
- ‘Rental’ – Rental (Internet)
- ‘EST’ – Electronic Sell Through (Internet)



## 5.2.11.2 DigitalAssetCardset-type

A cardset is a collection of cards for one purpose and displayed together. The reason it is a cardset may contain multiple individual cards. For example, a US anti-piracy cardset may contain an FBI card, a Department of Homeland Security card, and others.

Element	Attribute	Definition	Value	Card.
Cardset-type				
Type		Type of cardset. See below.	xs:string	1..n
Description		Description of cardset (human readable)	Xs:string	1..n
Sequence		Order of display for this cardset. A higher number represents later display. Cardsets with the same sequence must not overlap Region.	xs:positiveInteger	0..1

Type is encoded as follows:

- ‘AntiPiracy’ – Anti-piracy notices
- ‘DistributionLogo’ – Logos associated with distribution entity or entities.
- ‘Rating’ – Content Rating
- ‘DubbingCredit’ – Credits for dubbing (e.g., French talent dubbing in French)
- ‘Intermission’ – Information displayed during an intermission.
- ‘EditNotice’ – Information displayed regarding the edit (e.g., “This movie has been modified from the original version. It has been formatted to fit your screen.”)
- Other

## 6 CONTAINER METADATA

The Container Metadata describes the container that includes the various media pieces and the glue that holds them together.

### 6.1 Container Metadata Description

Logically speaking, the container holds a collection of tracks as described using `md:DigitalAssetMetadata-type`. The container packages these data in accordance with the rules for that container type, defined with the `md:ContainerType` element.

Often, the container type definition alone is not enough information to access the media in the container. `md:ContainerSpecificMetadata` may be included to provide any additional necessary information. Container-specific metadata definitions are not included in this version of the specification, so the `xs:any` type is used.

If `ContainerSpecificInformation` is provided, the `md:TrackRef` elements in the Digital Asset Metadata types may be used to cross reference. For example, container-specific metadata may map an MPEG-2 transport stream PID to a given Track.

### 6.2 Definitions

#### 6.2.1 ContainerMetadata-type

This type describes a container that in turn contains one or more audio, video, subtitle or image tracks.

Element	Attribute	Definition	Value	Card.
ContainerMetadata-type				
ContainerType		Identification of container type	<code>md:DigitalAssetContainerType-type</code>	0..1
Track		Track metadata	<code>md:ContainerTrackMetadata-type</code>	1..n
Hash		Hash of container. Multiple instances may be included if multiple methods are used.	<code>md:Hash-type</code>	0..n
Size		Size of container in bytes (octets).	<code>xs:positiveInteger</code>	0..1
ContainerSpecificMetadata		Additional information about the content and structure of the container. In the future, container-specific information will be provided.	<code>md:ContainerSpecific-type</code>	0..1
(any)		Any additional definitions	<code>xs:any##other</code>	0..n

---

#### 6.2.1.1 Container Type encoding, ContainerType-type

Container type is of simple type ContainerType-type that is xs:string. It may contain one of the following values:

- ‘3GP’ – Third Generation Partnership Project (3GPP) file format
- ‘3GP2’ – 3GPP2 file format
- ‘AC3’ – Dolby Digital file
- ‘AIFF’ – Audio Interchange File Format
- ‘ASF’ – Microsoft Advanced Streaming Format
- ‘AVI’ – Microsoft Audio Video Interleave, also includes AVI 2.0
- ‘CFF’ – Common File Format (UltraViolet)
- ‘DIVX’ – DivX movie file
- ‘DTS’ – DTS encoded file
- ‘FLV’ – Flash Video File
- ‘HCT’ – Hectavision File
- ‘ISO’ – ISO Container ISO/IEC 14496-12, when not specified in a more specific fashion (e.g, MP4)
- ‘JPEG’ – JPEG image file
- ‘M4V’ – Apple M4V
- ‘MJ2’ – JPEG 2000 file format; ‘ISO’ containing JPEG 2000
- ‘MP4’ – MPEG-4 Part 14, ISO/IEC 14496-14:2003
- ‘MKV’ – Matroska multimedia container
- ‘MPEG-2 (TS)’ – MPEG-2 Transport stream
- ‘MPEG-2 (PS)’ – MPEG-2 Program Stream
- ‘MXF’ – SMPTE MXF file
- ‘Ogg’ – Xiph.Org file format for Vorbis and Theora
- ‘Quicktime (MOV)’ – Apple QuickTime movie file
- ‘PNG’ – Portable Network Graphics (PNG) file
- ‘RIFF’ – Resource Interchange File Format
- ‘RM’ – RealNetwork’s RealMedia file format
- ‘SWF’ – Adobe Shockwave Flash
- ‘TIFF’ – tagged image file format
- ‘WMV’ – Microsoft WMV file
- ‘VOB’ – DVD Video Object file
- ‘XMF’ – XMF music file (MIDI)
- ‘ZIP’ – ZIP file
- ‘other’

If the format is not in this list, it is acceptable to include the Windows file extension. When using this form, precede with 'EXT:'. For example, 'EXT:DXR' for Macromedia Director Movie File (.dvr file extension).

Standard encoding is preferred and will be investigated.

### 6.2.1.2 ContainerSpecific-type

ContainerSpecific-type is a sequence of 0..n any##other. This allows any container metadata to be used here.

### 6.2.1.3 ContainerTrackMetadata-type

Element	Attribute	Definition	Value	Card.
ContainerTrackMetadata-type				
Audio		Metadata for an audio asset	md:DigitalAssetAudioData-type	(choice)
Video		Metadata for a video asset	md:DigitalAssetVideoData-type	(choice)
Subtitle		Metadata for subtitles	md:DigitalAssetSubtitleData-type	(choice)
Image		Metadata for Images	md:DigitalAssetImageData-type	(choice)
Interactive		Metadata for Interactive	md:DigitalAssetInteractiveData-type	(choice)
Container		Container encapsulated within the container (recursive).	md:ContainerMetadata-type	(choice)
ExternalTrackReference		Reference to a track that is external. It may be a standalone track or part of another container. If part of a container, the trackReference attribute should point to the track in the other container.	md:DigitalAssetExternalTrackReference-type	(choice)
InternalTrackReference		Reference to a track that is internal to the Container. This is used when it is preferred to refer to track by IDs rather than metadata.	xs:string	(choice)

### 6.2.2 ContainerProfile-type

ContainerProfile-type is defined as xs:string. It may be used to specify a profile for a given container. There are no enumerations currently defined.

---

## 7 CONTENT RATINGS

Common Metadata supports content advisory based on formal ratings systems along with an “Adult only” flag for non-rated adult material and to allow limited cross-system blocking of content.

### 7.1 Description

Ratings are of the form: Region/System/Rating/Reason. There is also type (e.g., Film, TV and Music) but this is generally subsumed by the System and implicit in the content (exceptions are handled).

### 7.2 Rules

There is no implied cross-mapping between advisory systems.

#### 7.2.1 “Unrated”

A system of “Unrated” can be created in any region, although it is not tied to any particular rating system. Within the system “Unrated” there are four ratings: “Unrated”, “higher”, “neutral” and “lower”. A rating of “Unrated” is a simple statement that the title is unrated in that region. The ratings “higher”, “neutral” and “lower” correspond with the condition encoding definitions below.

‘Unrated’ literally means that this particular media instance has not been rated. This frequently means that a work has never been self-rated or submitted to a ratings body, for example, because of the nature of the work (e.g., a sporting event) or for budgetary reasons.

‘Unrated’ is also used as a marketing term to reflect a work that contains additional material, generally implied as material that would change the rating, often represented something like, “*The Unrated Edition*”.

As a best practice, if the unrated work is derived from a rated work, the parent work should be included in the Parent element of the BasicMetadata-type with a relationshipType attribute of ‘isderivedfrom’. Although the content is still unrated, the recipient will have additional information on how they may wish to classify the work.

### 7.3 Definition

This section specifies the structure that can include a complete content rating set for a title.

#### 7.3.1 ContentRating-type

This element describes content-specific parental control information as provided by the content owner or rating agency.

NotRated and RatingsMatrix are an XSD ‘choice’. If NotRated is chosen, it must be ‘true’. NotRated is used if there are no other ratings.

The absence of a rating in a particular system does not necessarily imply the content is unrated. However, in most cases it can be assumed that it is unrated. Specifications based on Common Metadata should include a requirement that all relevant ratings be included if available.

Element	Attribute	Definition	Value	Card.
ContentRating-type				
NotRated		Has the content never been rated? 'true'=not rated. Must be 'true' if included.	xs:boolean	(choice)
	condition	An indication of the nature of the unrated status.	xs:string	0..1
Rating		Rating information	md:ContentRatingDetail-type	(choice) 1..n
AdultContent		Should content be blocked for all non-adult viewers? 'true'= yes. 'false' or absent means no. There is no formal definition of 'adult' content, and this represents the judgment of the originator.	xs:boolean	0..1

NotRated is distinguished from “unrated”. As mentioned above, the term “unrated” is often used as a marketing term. “unrated” may be used as a keyword to indicate this type of version.

### 7.3.1.1 Condition encoding

Condition is an indication of why the work is unrated. If condition is absent, no conclusions can be drawn regarding why work is unrated.

If the BasicMetadata-type element has a parent element and condition is included, the values shall be one of the following:

- ‘higher’ – The intent is for the work to have a rating higher than or equal to the parent’s rating. This is typically used for the “Unrated Edition” edits.
- ‘neutral’ –The work was not explicitly rated, but is intended to have the same rating as its parent object.
- ‘lower’ – The work is derived in such a way as to lower the rating. The intent is for the work to have a rating lower than or equal to the parent’s rating. This would be typical of TV or airplane edit. This would also apply to edits for particular cultural or religious sensitivities.
- ‘exempt’ – The work is exempt from ratings.

If the BasicMetadata-type element does not have a parent element and condition should not be included. This can be interpreted as “never rated.”

## 7.3.2 ContentRatingDetail-type

This element describes content-specific parental control information as provided by the content owner or rating agency.

Values come from Section 8, “Content Rating Encoding”. Values should be exactly as entered in the table in Section 8.

Element	Attribute	Definition	Value	Card.
ContentRatingDetail-type				
Region		Country/Region. Uses region encoding	md:Region-type	
System		Rating System	xs:string	
Value		Rating Value	xs:string	
Reason		Rating Reason. Only one Reason per element (i.e., either “L” or “V”, but not “LV”.)	xs:string	0..n
LinkToLogo		If there is an image associated with this rating, the link may be provided	xs:anyURI	0..1
Description		A string associated with the rating, such as, “ <b>Rated PG</b> For mild thematic elements and brief smoking”	xs:string	0..1

---

## 8 CONTENT RATING ENCODING

Encoding for content ratings has been moved to its own document, TR-META-CR found at <http://www.movie labs.com/md/ratings>. We recommend using the latest version of this document.



## 9 SELECTED EXAMPLES

Following are selected examples. These and other examples will appear on the web site.

### 9.1 People Name Examples

The following example was based on this test schema

```
<xs:element name="Person-name" type="md:PersonName-type"/>
<xs:element name="People">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="Person" type="md:BasicMetadataPeople-type" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

The following example covers the following people: Gorillaz, Kid n' Play, Cher, 50 Cent, MC Hammer, Dita von Teese, Marilyn Manson, Teenage Mutant Ninja Turtles, James van der Beek, Max von Sydow, Kat von D, Benjamin “Scatman” Crothers, and Peter Sellers. Note that Teenage Mutant Ninja Turtles is not a real entity and therefore will not be encoded, but it was included to test completeness.

```
<mdtest:People xsi:schemaLocation="http://www.movelabs.com/md/mdtest mdtest.xsd"
xmlns:md="http://www.movelabs.com/md" xmlns:mdtest="http://www.movelabs.com/md/mdtest"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <mdtest:Person>
    <md:Job>
      <md:JobFunction>Music Group</md:JobFunction>
      <md:JobDisplay>Band</md:JobDisplay>
    </md:Job>
    <md:Name>
      <md:DisplayName>Gorillaz</md:DisplayName>
      <md:SortName>Gorillaz</md:SortName>
    </md:Name>
    <md:Gender>neutral</md:Gender>
  </mdtest:Person>
  <mdtest:Person>
    <md:Job>
      <md:JobFunction>Other Group</md:JobFunction>
      <md:JobDisplay>Comedy Duo</md:JobDisplay>
    </md:Job>
    <md:Name>
      <md:DisplayName>Kid 'n Play</md:DisplayName>
      <md:SortName>Kid 'n Play</md:SortName>
    </md:Name>
    <md:Gender>male</md:Gender>
  </mdtest:Person>
```

```
<mdtest:Person>
  <md:Job>
    <md:JobFunction>Singer</md:JobFunction>
    <md:JobDisplay>Singer</md:JobDisplay>
  </md:Job>
  <md:Name>
    <md:DisplayName>Cher</md:DisplayName>
    <md:SortName>Cher</md:SortName>
  </md:Name>
  <md:Gender>female</md:Gender>
</mdtest:Person>
<mdtest:Person>
  <md:Job>
    <md:JobFunction>Singer</md:JobFunction>
    <md:JobDisplay>Rapper</md:JobDisplay>
  </md:Job>
  <md:Name>
    <md:DisplayName>50 Cent</md:DisplayName>
    <md:SortName>50 Cent</md:SortName>
    <md:FirstGivenName>Curtis</md:FirstGivenName>
    <md:SecondGivenName>James</md:SecondGivenName>
    <md:FamilyName>Jackson</md:FamilyName>
    <md:Suffix>III</md:Suffix>
  </md:Name>
  <md:Gender>male</md:Gender>
</mdtest:Person>
<mdtest:Person>
  <md:Job>
    <md:JobFunction>Singer</md:JobFunction>
    <md:JobDisplay>Rapper</md:JobDisplay>
  </md:Job>
  <md:Name>
    <md:DisplayName>MC Hammer</md:DisplayName>
    <md:SortName>MC Hammer</md:SortName>
  </md:Name>
  <md:Gender>male</md:Gender>
</mdtest:Person>
<mdtest:Person>
  <md:Job>
    <md:JobFunction>Dancer</md:JobFunction>
    <md:JobDisplay>Burlesque Dancer</md:JobDisplay>
  </md:Job>
  <md:Name>
    <md:DisplayName>Dita von Teese</md:DisplayName>
    <md:SortName>Von Teese, Dita</md:SortName>
    <md:FirstGivenName>Dita</md:FirstGivenName>
    <md:FamilyName>Von Teese</md:FamilyName>
  </md:Name>
  <md:Gender>female</md:Gender>
</mdtest:Person>
```

```
<mdtest:Person>
  <md:Job>
    <md:JobFunction>Singer</md:JobFunction>
    <md:JobDisplay>Singer</md:JobDisplay>
  </md:Job>
  <md:Name>
    <md:DisplayName>Marilyn Manson</md:DisplayName>
    <md:SortName>Manson, Marilyn</md:SortName>
    <md:FirstGivenName>Marilyn</md:FirstGivenName>
    <md:FamilyName>Manson</md:FamilyName>
  </md:Name>
  <md:Gender>male</md:Gender>
</mdtest:Person>
<mdtest:Person>
  <md:Job>
    <md:JobFunction>Other Group</md:JobFunction>
    <md:JobDisplay>Superhero Turtles</md:JobDisplay>
  </md:Job>
  <md:Name>
    <md:DisplayName>Teenage Mutant Ninja Turtles</md:DisplayName>
    <md:SortName>Teenage Mutant Ninja Turtles</md:SortName>
  </md:Name>
  <md:Gender>neutral</md:Gender>
</mdtest:Person>
<mdtest:Person>
  <md:Job>
    <md:JobFunction>Actor</md:JobFunction>
    <md:JobDisplay>Actor</md:JobDisplay>
    <md:BillingBlockOrder>1</md:BillingBlockOrder>
    <md:Character>Dawson Leery</md:Character>
  </md:Job>
  <md:Name>
    <md:DisplayName>James Van Der Beek</md:DisplayName>
    <md:SortName>Van Der Beek</md:SortName>
    <md:FirstGivenName>James</md:FirstGivenName>
    <md:SecondGivenName>William</md:SecondGivenName>
    <md:FamilyName>Van Der Beek</md:FamilyName>
    <md:Suffix>Jr.</md:Suffix>
  </md:Name>
  <md:Gender>male</md:Gender>
</mdtest:Person>
<mdtest:Person>
  <md:Job>
    <md:JobFunction>Actor</md:JobFunction>
    <md:JobDisplay>Actor</md:JobDisplay>
    <md:Character>Otto Frank</md:Character>
  </md:Job>
  <md:Name>
    <md:DisplayName>Max von Sydow</md:DisplayName>
    <md:SortName>von Sydow</md:SortName>
```

```

    <md:FirstGivenName>Max</md:FirstGivenName>
    <md:FamilyName>von Sydow</md:FamilyName>
  </md:Name>
  <md:Gender>male</md:Gender>
</mdtest:Person>
<mdtest:Person>
  <md:Job>
    <md:JobFunction>Artist/Performer</md:JobFunction>
    <md:JobDisplay>Tattoo Artist</md:JobDisplay>
  </md:Job>
  <md:Name>
    <md:DisplayName>Kat von D</md:DisplayName>
    <md:SortName>String</md:SortName>
    <md:FirstGivenName>Kat</md:FirstGivenName>
    <md:FamilyName>von D</md:FamilyName>
  </md:Name>
  <md:Gender>female</md:Gender>
</mdtest:Person>
<mdtest:Person>
  <md:Job>
    <md:JobFunction>Singer</md:JobFunction>
    <md:JobDisplay>Scat Singer</md:JobDisplay>
  </md:Job>
  <md:Name>
    <md:DisplayName>Scatman Crothers</md:DisplayName>
    <md:SortName>Scatman Crothers</md:SortName>
    <md:FirstGivenName>Benjamin</md:FirstGivenName>
    <md:SecondGivenName>Sherman</md:SecondGivenName>
    <md:FamilyName>Crothers</md:FamilyName>
    <md:Moniker>Scatman</md:Moniker>
  </md:Name>
  <md:Gender>male</md:Gender>
</mdtest:Person>
<mdtest:Person>
  <md:Job>
    <md:JobFunction>Actor</md:JobFunction>
    <md:JobDisplay>Actor</md:JobDisplay>
    <md:BillingBlockOrder>1</md:BillingBlockOrder>
    <md:Character>Group Captain Lionel Mandrake</md:Character>
    <md:Character>President Merkin Muffley</md:Character>
    <md:Character>Dr. Strangelove</md:Character>
  </md:Job>
  <md:Name>
    <md:DisplayName>Peter Sellers</md:DisplayName>
    <md:SortName>Sellers</md:SortName>
    <md:FirstGivenName>Peter</md:FirstGivenName>
    <md:FamilyName>Sellers</md:FamilyName>
  </md:Name>
  <md:Gender>male</md:Gender>
</mdtest:Person>

```

</mdtest:People>

## 9.2 Release History Example

The following example is based on this test schema:

```
<xs:element name="ReleaseHistorySet">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="ReleaseHistory" type="md:ReleaseHistory-type" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

The following history is included:

- US Theatrical: 2008-02-08
- US Fullscreen DVD: 2008-06-17
- US Widescreen DVD: 2008-06-17
- UK Theatrical: 2008-05-30
- UK DVD: 2008-09-22

```
<mdtest:ReleaseHistorySet xsi:schemaLocation="http://www.movielabs.com/md/mdtest mdtest.xsd"
  xmlns:md="http://www.movielabs.com/md" xmlns:mdtest="http://www.movielabs.com/md/mdtest"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <mdtest:ReleaseHistory>
    <md:ReleaseType>original</md:ReleaseType>
    <md:DistrTerritory>
      <md:country>US</md:country>
    </md:DistrTerritory>
    <md:Date>2008-02-08</md:Date>
    <md:Description>US Theatrical Release</md:Description>
  </mdtest:ReleaseHistory>
  <mdtest:ReleaseHistory>
    <md:ReleaseType>DVD</md:ReleaseType>
    <md:DistrTerritory>
      <md:country>US</md:country>
    </md:DistrTerritory>
    <md:Date>2008-06-17</md:Date>
    <md:Description>US Fullscreen Edition</md:Description>
  </mdtest:ReleaseHistory>
  <mdtest:ReleaseHistory>
    <md:ReleaseType>DVD</md:ReleaseType>
    <md:DistrTerritory>
      <md:country>US</md:country>
    </md:DistrTerritory>
```

```

<md:Date>2008-06-17</md:Date>
<md:Description>US Widescreen Edition</md:Description>
</mdtest:ReleaseHistory>
<mdtest:ReleaseHistory>
  <md:ReleaseType>original</md:ReleaseType>
  <md:DistrTerritory>
    <md:country>GB</md:country>
  </md:DistrTerritory>
  <md:Date>2008-05-30</md:Date>
  <md:Description>UK Theatrical Release</md:Description>
</mdtest:ReleaseHistory>
<mdtest:ReleaseHistory>
  <md:ReleaseType>DVD</md:ReleaseType>
  <md:DistrTerritory>
    <md:country>GB</md:country>
  </md:DistrTerritory>
  <md:Date>2008-09-22</md:Date>
  <md:Description>UK Release</md:Description>
</mdtest:ReleaseHistory>
</mdtest:ReleaseHistorySet>

```

## 9.3 Content Rating Examples

The following example was based on this test schema:

```
<xs:element name="RatingSet" type="md:ContentRating-type"/>
```

The following ratings are given:

- US, MPAA, PG-13
- UK, BBFC, 12
- US, TV Parental Guidelines, TV14, coarse or crude language, sexual situations and violence
- Canada/Ontario, OFRB, 14A

```

<mdtest:RatingSet xsi:schemaLocation="http://www.movielabs.com/md/mdtest mdtest.xsd"
xmlns:md="http://www.movielabs.com/md" xmlns:mdtest="http://www.movielabs.com/md/mdtest"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <md:Rating>
    <md:Region>
      <md:country>US</md:country>
    </md:Region>
    <md:System>MPAA</md:System>
    <md:Value>PG-13</md:Value>
    <md:LinkToLogo>http://www.mpa.org/_images/parental-strongly.gif</md:LinkToLogo>
  </md:Rating>

```

```
<md:Rating>
  <md:Region>
    <md:country>GB</md:country>
  </md:Region>
  <md:System>BBFC</md:System>
  <md:Value>12</md:Value>
  <md:LinkToLogo>http://www.bbfc.co.uk/images/classification/c-12.gif</md:LinkToLogo>
</md:Rating>
<md:Rating>
  <md:Region>
    <md:country>US</md:country>
  </md:Region>
  <md:System>TVPG</md:System>
  <md:Value>TV14</md:Value>
  <md:Reason>L</md:Reason>
  <md:Reason>S</md:Reason>
  <md:Reason>V</md:Reason>
  <md:LinkToLogo>http://www.tvguidelines.org/images/tv14.jpg</md:LinkToLogo>
</md:Rating>
<md:Rating>
  <md:Region>
    <md:countryRegion>CA-ON</md:countryRegion>
  </md:Region>
  <md:System>OFRB</md:System>
  <md:Value>14A</md:Value>
  <md:LinkToLogo>http://www.ofrb.gov.on.ca/english/images/14a\_high.gif</md:LinkToLogo>
</md:Rating>
</mdtest:RatingSet>
```

## 10 REDEFINE SUPPORT

It is anticipated that schemas that use Common Metadata will wish to control vocabularies or otherwise constrain the schema. This is fully allowed.

Note that unless you intend to do redefines, this section is internal to the schema and has no effect on schema users.

To support controlled vocabularies, Common Metadata provides a set of simple types corresponding with particular elements and attributes. These are of the form

`<XML type>-<reference>`

Where

- `<XML type>` refers to the original XML type of the attribute or element. For example, `xs:string` elements start with 'string'.
- `<reference>` is a descriptive term relating to the attribute or element in question. For example, `BasicMetadata/Genre`'s reference is 'Genre'.

### 10.1 General XML Type Redefines

These redefines are used across all uses of the XML type.

XML type	Redefine type
<code>xs:language</code>	<code>Md:language-redefine</code>

### 10.2 Type-specific Redefines

The following tables list the element or attribute that is subject to redefine and the simple type that redefines that value. For example, for the element `//PersonName/Suffix`, there is a simple type `md:string-Name-Suffix` that can be redefined to control the `Suffix` element's pattern or enumeration.

The Contains enumerations column indicates whether the 'Redefine type' already includes enumerations. In that case, the only constraining option available, according to XML redefine rules, is to restrict out one or more of those enumerations.

Note that Common Metadata suggests many vocabularies that are not enforced by XML enumerations.

#### 10.2.1 Identifiers

The following applies to identifiers. This is applicable when only specific identifiers are allowed. If only one Namespace is allowed, one might wish to define Identifier as a pattern.



Element or Attribute	Redefine type	Contains enumerations
ContentIdentifier-type/Namespace	md:string-ContentID-Namespace	
ContentIdentifier-type/Identifier	md:string-ContentID-Identifier	

## 10.2.2 Basic Metadata

Element or Attribute	Redefine type	Contains enumerations
//BasicMetadataInfo-type /ArtReference/@resolution	md:string-ArtReference-resolution	
//BasicMetadataInfo-type /DisplayIndicators	md:string-DisplayIndicators	yes
//BasicMetadataInfo-type /Genre	md:string-Genre	
//BasicMetadataInfo-type /Genre/@id	md:string-Genre_id	
//BasicMetadataInfo-type /Keyword	md:string-Keyword	
//BasicMetadataInfo-type/TitleAlternate	md:string-TitleAlternate_type	
//BasicMetadataJob-type/JobFunction	md:string-JobFunction	
//BasicMetadataJob-type/JobFunction/@scheme	md:string-JobFunction-scheme	
//BasicMetadataJob-type/JobDisplay	md:string-JobDisplay	
//BasicMetadata/WorkType	md:string-WorkType	
//BasicMetadata/WorkTypeDetail	md:string-WorkTypeDetail	
//BasicMetadata/PictureFormat	md:string-PictureFormat	
//BasicMetadata/AspectRatio	md:string-AspectRatio	
//BasicMetadata/AssociatedOrg/@role	md:string-AssociatedOrg-role	

## 10.2.3 Digital Asset Metadata

Element or Attribute	Redefine type	Contains enumerations
//DigitalAssetAudio-type/Type	md:string-Audio-Type	
//DigitalAssetAudio-type/Language	md:DigitalAssetAudioLanguage-type	
//DigitalAssetAudio-type/Channels	md:string-Audio-Channels	
//DigitalAssetAudio-type/TrackReference	md:string-TrackReference <sup>1</sup>	
//DigitalAssetAudioEncoding-type/Codec	md:string-Audio-Enc-Codec	
//DigitalAssetAudioEncoding-type/CodecType	md:string-Audio-Enc-CodecType	
//DigitalAssetAudioEncoding-type/ChannelMapping	md:string-Audio-Enc-ChannelMapping	
//DigitalAssetVideo-type/Type	md:string-Video-Type	
//DigitalAssetVideo-type/PictureFormat	md:string-Video-PictureFormat	
//DigitalAssetVideo-type/SubtitleLanguage	md:DigitalAssetVideoSubtitleLanguage-type	
//DigitalAssetVideo-type/TrackReference	md:string-TrackReference <sup>1</sup>	
//DigitalAssetVideoEncoding-type/Codec	md:string-Video-Enc-Codec	
//DigitalAssetVideoEncoding-type/CodecType	md:string-Video-Enc-CodecType	
//DigitalAssetVideoEncoding-type/MPEGProfile	md:string-Video-Enc-MProfile	
//DigitalAssetVideoEncoding-type/MPEGLevel	md:string-Video-Enc-MLevel	
//DigitalAssetVideoEncoding-type/VBR	md:string-Video-Enc-VBR	
//DigitalAssetVideoPicture-type/AspectRatio	md:string-Video-Pic-AspectRatio	Yes
//DigitalAssetVideoPicture-type/PixelAspect	md:string-Video-Pic-PixelAspect	
//DigitalAssetVideoPicture-type/ColorSampling	md:string-Video-Pic-ColorSampling	
//DigitalAssetVideoPicture-type/Colorimetry	md:string-Video-Pic-Colorimetry	
//DigitalAssetVideoPicture-type/FrameRate/@multiplier	md:string-Video-Pic-FrameRate-multiplier	Yes

Element or Attribute	Redefine type	Contains enumerations
//DigitalAssetVideoPicture-type/FrameRate/@timecode	md:string-Video-Pic-FrameRate-timecode	
//DigitalAssetVideoPicture-type/Progressive/@scanOrder	md:string-Video-Pic-Progressive-scanOrder	Yes
//DigitalAssetVideoPicture-type/Type3D	md:string-Video-Pic-Type3D	
//DigitalAssetSubtitle-type/Format	md:DigitalAssetSubtitleFormat-type md:string-Subtitle-Format	
//DigitalAssetSubtitle-type/Type	md:string-Subtitle-Type	
//DigitalAssetSubtitle-type/FormatType	md:string-Subtitle-FormatType	
//DigitalAssetImage-type/Encoding	md:string-Image-Encoding	
//DigitalAssetImage-type/TrackReference	md:string-TrackReference <sup>1</sup>	
//DigitalAssetInteractiveData-type/Type	md:string-Interactive-Type	
//DigitalAssetInteractiveData-type/FormatType	md:string-Interactive-FormatType	
//DigitalAssetInteractiveEncoding-type/RuntimeEnvironment	md:string-Interactive-Enc- RuntimeEnvironment	
//DigitalAssetInteractiveEncoding-type/FirstVersion	md:string-Interactive-Enc-Version <sup>1</sup>	
//DigitalAssetInteractiveEncoding-type/FirstVersion	md:string-Interactive-Enc-Version <sup>1</sup>	
//DigitalAssetInteractive-type/TrackReference	md:string-TrackReference <sup>1</sup>	
//DigitalAssetCardsetList-type/Location	md:string-CardsetList-Location	
//DigitalAssetCardset-type/Type	md:string-Cardset-Type	
//DigitalAssetWatermark-type/Vendor	md:string-Watermark_Vendor	

<sup>1</sup> This type is used for more than one element or attribute.

## 10.2.4 Content Ratings

Element or Attribute	Redefine type	Contains enumerations
//ContentRating-type/NotRated/@condition	md:string-NotRated-condition	
//ContentRatingDetail-type/System	md:string-Rating-System	

## 10.2.5 Container Metadata

Element or Attribute	Redefine type	Contains enumerations
ContainerMetadata-type/Type	md:string-Container-Type	

## 10.2.6 Compilation Object

Element or Attribute	Redefine type	Contains enumerations
CompObj-type/EntryNumber	md:string-Compilation-EntryNumber	
CompObj-type/EntryClass	md:string-Compilation-EntryClass	
CompObj-type/CompilationClass	md:CompObjClass-type md:string-CompilationClass	

### 10.2.7 Additional Types

Element or Attribute	Redefine type	Contains enumerations
//PersonName-type/Suffix	md:string-Name-Suffix	
//PersonIdentifier-type/Namespace	md:string-Identifier-Namespace	
//ReleaseHistory-type/ReleaseType	md:string-ReleaseType	
//ReleaseHistory-type/ReleaseOrg/@idType	md:string-ReleaseOrg-idType	
//Money-type/@currency	md:string-Money-currency	
//Hash	md:string-Hash	
//Hash/@method	md:string-Hash-method	

### 10.2.8 Release History

Element or Attribute	Redefine type	Contains enumerations
//ReleaseHistory-type/ReleaseType	md:string-Release-ReleaseType	



This XML file does not appear to have any style information associated with it. The document tree is shown below.

---

```
<!--
  edited with XMLSpy v2013 sp1 (x64) (http://www.altova.com) by Craig Seidel
  (MovieLabs)
-->
<xs:schema xmlns:md="http://www.movielabs.com/schema/md/v2.0/md"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://www.movielabs.com/schema/md/v2.0/md"
elementFormDefault="qualified" attributeFormDefault="unqualified" version="2.0">
  <!-- Common Metadata -->
  <!-- Verson 2.0 -->
  <!-- *** Metadata References *** -->
  <xs:simpleType name="id-type">
    <xs:restriction base="xs:anyURI"/>
  </xs:simpleType>
  <xs:simpleType name="orgID-type">
    <xs:restriction base="xs:string"/>
  </xs:simpleType>
  <xs:simpleType name="ContentID-type">
    <xs:restriction base="md:id-type"/>
  </xs:simpleType>
  <xs:complexType name="ContentIdentifier-type">
    <xs:sequence>
      <xs:element name="Namespace" type="md:string-ContentID-Namespace"/>
      <xs:element name="Identifier" type="md:string-ContentID-Identifier"/>
      <xs:element name="Location" type="xs:anyURI" minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="ContentNumber-type">
    <xs:simpleContent>
      <xs:extension base="xs:string">
        <xs:attribute name="domain"/>
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>
  <xs:simpleType name="AssetPhysicalID-type">
    <xs:restriction base="md:id-type"/>
  </xs:simpleType>
  <xs:simpleType name="AssetLogicalID-type">
    <xs:restriction base="md:id-type"/>
  </xs:simpleType>
  <!-- *** Parental Controls *** -->
  <xs:complexType name="ContentRatingDetail-type">
    <xs:sequence>
      <xs:element name="Region" type="md:Region-type"/>
      <xs:element name="System" type="md:string-Rating-System"/>
      <xs:element name="Value" type="xs:string"/>
      <xs:element name="Reason" type="xs:string" minOccurs="0"
maxOccurs="unbounded"/>
      <xs:element name="LinkToLogo" type="xs:anyURI" minOccurs="0"/>
      <xs:element name="Description" type="xs:string" minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>
```

```

<xs:complexType name="ContentRating-type">
  <xs:sequence>
    <xs:choice>
      <xs:element name="NotRated" fixed="true">
        <xs:complexType>
          <xs:simpleContent>
            <xs:extension base="xs:boolean">
              <xs:attribute name="condition" type="md:string-NotRated-condition"/>
            </xs:extension>
          </xs:simpleContent>
        </xs:complexType>
      </xs:element>
      <xs:element name="Rating" type="md:ContentRatingDetail-type"
        maxOccurs="unbounded"/>
    </xs:choice>
    <xs:element name="AdultContent" type="xs:boolean" minOccurs="0"/>
  </xs:sequence>
</xs:complexType>
<!-- *** Basic Metadata *** -->
<xs:complexType name="ContentSequenceInfo-type">
  <xs:sequence>
    <xs:element name="Number" type="xs:int" minOccurs="0"/>
    <xs:element name="DistributionNumber" type="md:ContentNumber-type"
      minOccurs="0"/>
    <xs:element name="HouseSequence" minOccurs="0">
      <xs:complexType>
        <xs:complexContent>
          <xs:extension base="md:ContentNumber-type"/>
        </xs:complexContent>
      </xs:complexType>
    </xs:element>
    <xs:element name="AlternateNumber" type="md:ContentNumber-type" minOccurs="0"
      maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="BasicMetadataParent-type">
  <xs:choice>
    <xs:element name="Parent" type="md:BasicMetadata-type"/>
    <xs:element name="ParentContentID" type="md:ContentID-type"/>
  </xs:choice>
  <xs:attribute name="relationshipType">
    <xs:simpleType>
      <xs:restriction base="xs:string">
        <xs:enumeration value="isclipof"/>
        <xs:enumeration value="isepisodeof"/>
        <xs:enumeration value="isseasonof"/>
        <xs:enumeration value="ispieceof"/>
        <xs:enumeration value="ispartof"/>
        <xs:enumeration value="isderivedfrom"/>
        <xs:enumeration value="iscompositeof"/>
        <xs:enumeration value="issupplementto"/>
        <xs:enumeration value="ispromotionfor"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:attribute>
</xs:complexType>
<xs:complexType name="BasicMetadataInfo-type">

```



```

<xs:sequence>
  <xs:element name="TitleDisplay19" type="xs:string" minOccurs="0"/>
  <xs:element name="TitleDisplay60" type="xs:string" minOccurs="0"/>
  <xs:element name="TitleDisplayUnlimited" type="xs:string" minOccurs="0"/>
  <xs:element name="TitleSort" type="xs:string"/>
  <xs:element name="ArtReference" minOccurs="0" maxOccurs="unbounded">
    <xs:complexType>
      <xs:simpleContent>
        <xs:extension base="xs:anyURI">
          <xs:attribute name="resolution" type="md:string-
            ArtReference_resolution"/>
        </xs:extension>
      </xs:simpleContent>
    </xs:complexType>
  </xs:element>
  <xs:element name="Summary190">
    <xs:complexType>
      <xs:simpleContent>
        <xs:extension base="xs:string">
          <xs:attribute name="cast" type="xs:boolean"/>
        </xs:extension>
      </xs:simpleContent>
    </xs:complexType>
  </xs:element>
  <xs:element name="Summary400" minOccurs="0">
    <xs:complexType>
      <xs:simpleContent>
        <xs:extension base="xs:string">
          <xs:attribute name="cast" type="xs:boolean"/>
        </xs:extension>
      </xs:simpleContent>
    </xs:complexType>
  </xs:element>
  <xs:element name="Summary4000" minOccurs="0">
    <xs:complexType>
      <xs:simpleContent>
        <xs:extension base="xs:string">
          <xs:attribute name="cast" type="xs:boolean"/>
        </xs:extension>
      </xs:simpleContent>
    </xs:complexType>
  </xs:element>
  <xs:element name="DisplayIndicators" type="md:string-DisplayIndicators"
minOccurs="0" maxOccurs="unbounded"/>
  <xs:element name="Genre" minOccurs="0" maxOccurs="unbounded">
    <xs:complexType>
      <xs:simpleContent>
        <xs:extension base="xs:string">
          <xs:attribute name="source" type="xs:anyURI"/>
          <xs:attribute name="id" type="md:string-Genre_id"/>
          <xs:attribute name="level" type="xs:integer"/>
        </xs:extension>
      </xs:simpleContent>
    </xs:complexType>
  </xs:element>
  <xs:element name="Keyword" type="md:string-Keyword" minOccurs="0"
maxOccurs="unbounded"/>

```

```

<xs:element name="VersionNotes" type="xs:string" minOccurs="0"/>
<xs:element name="Region" type="md:Region-type" minOccurs="0"
maxOccurs="unbounded"/>
<xs:element name="OriginalTitle" type="xs:string" minOccurs="0"/>
<xs:element name="CopyrightLine" type="xs:string" minOccurs="0"/>
<xs:element name="PeopleLocal" type="md:BasicMetadataPeople-type" minOccurs="0"
maxOccurs="unbounded"/>
<xs:element name="TitleAlternate" minOccurs="0" maxOccurs="unbounded">
  <xs:complexType>
    <xs:simpleContent>
      <xs:extension base="xs:string">
        <xs:attribute name="type" type="md:string-TitleAlternate_type"/>
        <xs:attribute name="language" type="xs:language"/>
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>
</xs:element>
</xs:sequence>
<xs:attribute name="language" type="md:language-redefine" use="required"/>
<xs:attribute name="default" type="xs:boolean"/>
</xs:complexType>
<xs:complexType name="BasicMetadataJob-type">
  <xs:sequence>
    <xs:element name="JobFunction">
      <xs:complexType>
        <xs:simpleContent>
          <xs:extension base="md:Role-type">
            <xs:attribute name="scheme" type="md:string-JobFunction_scheme"/>
          </xs:extension>
        </xs:simpleContent>
      </xs:complexType>
    </xs:element>
    <xs:element name="JobDisplay" minOccurs="0" maxOccurs="unbounded">
      <xs:complexType>
        <xs:simpleContent>
          <xs:extension base="md:string-JobDisplay">
            <xs:attribute name="language" type="xs:language"/>
          </xs:extension>
        </xs:simpleContent>
      </xs:complexType>
    </xs:element>
    <xs:element name="BillingBlockOrder" type="xs:int" minOccurs="0"/>
    <xs:element name="Character" type="xs:string" minOccurs="0"
maxOccurs="unbounded"/>
    <xs:element name="Guest" type="xs:boolean" minOccurs="0"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="BasicMetadataPeople-type">
  <xs:sequence>
    <xs:element name="Job" type="md:BasicMetadataJob-type" maxOccurs="unbounded"/>
    <xs:element name="Name" type="md:PersonName-type"/>
    <xs:element name="Identifier" type="md:PersonIdentifier-type" minOccurs="0"
maxOccurs="unbounded"/>
    <xs:element name="Gender" minOccurs="0">
      <xs:simpleType>
        <xs:restriction base="xs:string">
          <xs:enumeration value="male"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:element>
  </xs:sequence>
</xs:complexType>

```

```

        <xs:enumeration value="female"/>
        <xs:enumeration value="neutral"/>
        <xs:enumeration value="plural"/>
    </xs:restriction>
</xs:simpleType>
</xs:element>
</xs:sequence>
</xs:complexType>
<xs:complexType name="BasicMetadata-type">
    <xs:sequence>
        <xs:element name="UpdateNum" minOccurs="0">
            <xs:simpleType>
                <xs:restriction base="xs:int">
                    <xs:minInclusive value="1"/>
                </xs:restriction>
            </xs:simpleType>
        </xs:element>
        <xs:element name="LocalizedInfo" type="md:BasicMetadataInfo-type"
maxOccurs="unbounded"/>
        <xs:element name="RunLength" type="xs:duration" minOccurs="0"/>
        <xs:element name="ReleaseYear" type="xs:gYear"/>
        <xs:element name="ReleaseDate" type="md:YearDateOrTime-type" minOccurs="0"/>
        <xs:element name="ReleaseHistory" type="md:ReleaseHistory-type" minOccurs="0"
maxOccurs="unbounded"/>
        <xs:element name="WorkType" type="md:string-WorkType"/>
        <xs:element name="WorkTypeDetail" type="md:string-WorkTypeDetail"
minOccurs="0"/>
        <xs:element name="PictureColorType" type="md:ColorType-type" minOccurs="0"/>
        <xs:element name="PictureFormat" type="md:string-PictureFormat" minOccurs="0"/>
        <xs:element name="ThreeD" type="xs:boolean" minOccurs="0"/>
        <xs:element name="AspectRatio" type="md:string-AspectRatio" minOccurs="0"/>
        <xs:element name="AltIdentifier" type="md:ContentIdentifier-type" minOccurs="0"
maxOccurs="unbounded"/>
        <xs:element name="RatingSet" type="md:ContentRating-type" minOccurs="0"/>
        <xs:element name="People" type="md:BasicMetadataPeople-type" minOccurs="0"
maxOccurs="unbounded"/>
        <xs:element name="CountryOfOrigin" type="md:Region-type" minOccurs="0"/>
        <xs:element name="PrimarySpokenLanguage" type="md:language-redefine"
minOccurs="0" maxOccurs="unbounded"/>
        <xs:element name="OriginalLanguage" type="md:language-redefine" minOccurs="0"
maxOccurs="unbounded"/>
        <xs:element name="VersionLanguage" type="md:language-redefine" minOccurs="0"
maxOccurs="unbounded"/>
        <xs:element name="AssociatedOrg" minOccurs="0" maxOccurs="unbounded">
            <xs:complexType>
                <xs:complexContent>
                    <xs:extension base="md:OrgName-type">
                        <xs:attribute name="role" type="md:string-AssociatedOrg_role"/>
                    </xs:extension>
                </xs:complexContent>
            </xs:complexType>
        </xs:element>
        <xs:element name="SequenceInfo" type="md:ContentSequenceInfo-type"
minOccurs="0"/>
        <xs:element name="Parent" type="md:BasicMetadataParent-type" minOccurs="0"
maxOccurs="unbounded"/>
    </xs:sequence>

```

```

    <xs:attribute name="ContentID" type="md:ContentID-type" use="required"/>
</xs:complexType>
<!-- *** Abbreviated Metadata *** -->
<xs:complexType name="AbbreviatedMetadataInfo-type">
  <xs:sequence>
    <xs:element name="TitleBrief" type="xs:string"/>
    <xs:element name="ArtReference" type="xs:anyURI" minOccurs="0"
maxOccurs="unbounded"/>
    <xs:element name="SummaryShort" type="xs:string"/>
    <xs:element name="DisplayIndicators" minOccurs="0" maxOccurs="unbounded">
      <xs:simpleType>
        <xs:restriction base="xs:string">
          <xs:enumeration value="CC"/>
          <xs:enumeration value="F"/>
          <xs:enumeration value="P"/>
          <xs:enumeration value="DD"/>
          <xs:enumeration value="SAP"/>
          <xs:enumeration value="DVS"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:element>
  </xs:sequence>
  <xs:attribute name="language" type="xs:language" use="required"/>
  <xs:attribute name="default" type="xs:boolean"/>
</xs:complexType>
<xs:complexType name="AbbreviatedMetadata-type">
  <xs:sequence>
    <xs:element name="UpdateNum" minOccurs="0">
      <xs:simpleType>
        <xs:restriction base="xs:int">
          <xs:minInclusive value="1"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:element>
    <xs:element name="LocalizedInfo" type="md:AbbreviatedMetadataInfo-type"
maxOccurs="unbounded"/>
    <xs:element name="Rating" type="md:ContentRating-type" minOccurs="0"
maxOccurs="unbounded"/>
    <xs:element name="AltIdentifier" type="md:ContentIdentifier-type" minOccurs="0"
maxOccurs="unbounded"/>
    <xs:element name="Studio" type="xs:string">
      <xs:annotation>
        <xs:documentation>Equivalent to DisplayName</xs:documentation>
      </xs:annotation>
    </xs:element>
  </xs:sequence>
  <xs:attribute name="ContentID" type="md:ContentID-type" use="required"/>
</xs:complexType>
<!-- *** Digital Asset Metadata *** -->
<xs:complexType name="DigitalAssetExternalTrackReference-type">
  <xs:simpleContent>
    <xs:extension base="md:ContentID-type">
      <xs:attribute name="namespace" type="xs:string"/>
      <xs:attribute name="location" type="xs:string"/>
      <xs:attribute name="trackReference" type="md:string-TrackReference"/>
    </xs:extension>
  </xs:simpleContent>

```

```

</xs:complexType>
<xs:complexType name="DigitalAssetCardset-type">
  <xs:sequence>
    <xs:element name="Type" type="md:string-Cardset-Type" maxOccurs="unbounded"/>
    <xs:element name="Description" type="xs:string" minOccurs="0"/>
    <xs:element name="Sequence" type="xs:positiveInteger" minOccurs="0"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="DigitalAssetCardsetList-type">
  <xs:sequence>
    <xs:element name="Type" type="md:string-CardsetList-Type" minOccurs="0"
      maxOccurs="unbounded"/>
    <xs:element name="Region" type="md:MadeForRegion-type" minOccurs="0"
      maxOccurs="unbounded"/>
    <xs:element name="Cardset" type="md:DigitalAssetCardset-type"
      maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="DigitalAssetWatermark-type">
  <xs:sequence>
    <xs:element name="Vendor" type="md:string-Watermark_Vendor"/>
    <xs:element name="ProductAndVersionID" type="xs:string"/>
    <xs:element name="Data" type="xs:string" minOccurs="0"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="DigitalAssetAudioEncoding-type">
  <xs:sequence>
    <xs:element name="Codec" type="md:string-Audio-Enc-Codec"/>
    <xs:element name="CodecType" type="md:string-Audio-Enc-CodecType" minOccurs="0"
      maxOccurs="unbounded"/>
    <xs:element name="BitrateMax" type="xs:integer" minOccurs="0"/>
    <xs:element name="BitrateAverage" type="xs:integer" minOccurs="0"/>
    <xs:element name="VBR" type="md:string-Audio-Enc-VBR" minOccurs="0"/>
    <xs:element name="SampleRate" type="xs:integer" minOccurs="0"/>
    <xs:element name="SampleBitDepth" type="xs:integer" minOccurs="0"/>
    <xs:element name="ChannelMapping" type="md:string-Audio-Enc-ChannelMapping"
      minOccurs="0"/>
    <xs:element name="Watermark" type="md:DigitalAssetWatermark-type" minOccurs="0"
      maxOccurs="unbounded"/>
    <xs:element name="ActualLength" type="xs:duration" minOccurs="0"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="DigitalAssetAudioLanguage-type">
  <xs:simpleContent>
    <xs:extension base="md:language-redefine">
      <xs:attribute name="dubbed" type="xs:boolean"/>
    </xs:extension>
  </xs:simpleContent>
</xs:complexType>
<xs:complexType name="DigitalAssetAudioData-type">
  <xs:sequence>
    <xs:element name="Description" type="xs:string" minOccurs="0"/>
    <xs:element name="Type" type="md:string-Audio-Type" minOccurs="0"/>
    <xs:element name="Encoding" type="md:DigitalAssetAudioEncoding-type"
      minOccurs="0"/>
    <xs:element name="Language" type="md:DigitalAssetAudioLanguage-type"/>
    <xs:element name="Channels" type="xs:string" minOccurs="0"/>
  </xs:sequence>

```

```

<xs:element name="TrackReference" type="md:string-TrackReference"
minOccurs="0"/>
<xs:element name="TrackIdentifier" type="md:ContentIdentifier-type"
minOccurs="0" maxOccurs="unbounded"/>
<xs:element name="Private" type="md:PrivateData-type" minOccurs="0"/>
</xs:sequence>
</xs:complexType>
<xs:complexType name="DigitalAssetVideoEncoding-type">
<xs:sequence>
<xs:element name="Codec" type="md:string-Video-Enc-Codec"/>
<xs:element name="CodecType" type="md:string-Video-Enc-CodecType" minOccurs="0"
maxOccurs="unbounded"/>
<xs:element name="MPEGProfile" type="md:string-Video-Enc-MProfile"
minOccurs="0"/>
<xs:element name="MPEGLLevel" type="md:string-Video-Enc-MProfile"
minOccurs="0"/>
<xs:element name="BitrateMax" type="xs:integer" minOccurs="0"/>
<xs:element name="BitRateAverage" type="xs:integer" minOccurs="0"/>
<xs:element name="VBR" type="md:string-Video-Enc-VBR" minOccurs="0"/>
<xs:element name="Watermark" type="md:DigitalAssetWatermark-type" minOccurs="0"
maxOccurs="unbounded"/>
<xs:element name="ActualLength" type="xs:duration" minOccurs="0"/>
</xs:sequence>
</xs:complexType>
<xs:complexType name="DigitalAssetVideoPicture-type">
<xs:sequence>
<xs:element name="AspectRatio" type="md:string-Video-Pic-AspectRatio"
minOccurs="0"/>
<xs:element name="PixelAspect" type="md:string-Video-Pic-PixelAspect"
minOccurs="0"/>
<xs:element name="WidthPixels" type="xs:int" minOccurs="0"/>
<xs:element name="HeightPixels" type="xs:int" minOccurs="0"/>
<xs:element name="ActiveWidthPixels" type="xs:int" minOccurs="0"/>
<xs:element name="ActiveHeightPixels" type="xs:int" minOccurs="0"/>
<xs:element name="FrameRate" minOccurs="0">
<xs:complexType>
<xs:simpleContent>
<xs:extension base="xs:int">
<xs:attribute name="multiplier" type="md:string-Video-Pic-FrameRate-
multiplier"/>
<xs:attribute name="timecode" type="md:string-Video-Pic-FrameRate-
timecode"/>
</xs:extension>
</xs:simpleContent>
</xs:complexType>
</xs:element>
<xs:element name="Progressive" minOccurs="0">
<xs:complexType>
<xs:simpleContent>
<xs:extension base="xs:boolean">
<xs:attribute name="scanOrder" type="md:string-Video-Pic-Progressive-
scanOrder"/>
</xs:extension>
</xs:simpleContent>
</xs:complexType>
</xs:element>
<xs:element name="ColorSubsampling" type="md:string-Video-Pic-ColorSampling"

```



```

    minOccurs="0"/>
    <xs:element name="Colorimetry" type="md:string-Video-Pic-Colorimetry"
    minOccurs="0"/>
    <xs:element name="Type3D" type="md:string-Video-Pic-Type3D" minOccurs="0"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="DigitalAssetVideoSubtitleLanguage-type">
  <xs:simpleContent>
    <xs:extension base="md:language-redefine">
      <xs:attribute name="closed" type="xs:boolean"/>
      <xs:attribute name="type" type="md:string-Video-SubLang-type"/>
    </xs:extension>
  </xs:simpleContent>
</xs:complexType>
<xs:complexType name="DigitalAssetVideoData-type">
  <xs:sequence>
    <xs:element name="Description" type="xs:string" minOccurs="0"/>
    <xs:element name="Type" type="md:string-Video-Type" minOccurs="0"/>
    <xs:element name="Encoding" type="md:DigitalAssetVideoEncoding-type"
    minOccurs="0"/>
    <xs:element name="Picture" type="md:DigitalAssetVideoPicture-type"/>
    <xs:element name="ColorType" type="md:ColorType-type" minOccurs="0">
      <xs:annotation>
        <xs:documentation>BW, Color, Colorized, etc.</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="PictureFormat" type="md:string-Video-PictureFormat"
    minOccurs="0"/>
    <xs:element name="SubtitleLanguage" type="md:DigitalAssetVideoSubtitleLanguage-
    type" minOccurs="0" maxOccurs="unbounded"/>
    <xs:element name="SignedLanguage" type="md:language-redefine" minOccurs="0"/>
    <xs:element name="CardsetList" type="md:DigitalAssetCardsetList-type"
    minOccurs="0" maxOccurs="unbounded"/>
    <xs:element name="TrackReference" type="md:string-TrackReference"
    minOccurs="0"/>
    <xs:element name="TrackIdentifier" type="md:ContentIdentifier-type"
    minOccurs="0" maxOccurs="unbounded"/>
    <xs:element name="Private" type="md:PrivateData-type" minOccurs="0"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="DigitalAssetImageData-type">
  <xs:sequence>
    <xs:element name="Width" type="xs:int"/>
    <xs:element name="Height" type="xs:int"/>
    <xs:element name="Encoding" type="md:string-Image-Encoding"/>
    <xs:element name="Language" type="md:language-redefine" minOccurs="0"/>
    <xs:element name="TrackReference" type="md:string-TrackReference"
    minOccurs="0"/>
    <xs:element name="TrackIdentifier" type="md:ContentIdentifier-type"
    minOccurs="0" maxOccurs="unbounded"/>
    <xs:element name="Private" type="md:PrivateData-type" minOccurs="0"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="DigitalAssetSubtitleFormat-type">
  <xs:simpleContent>
    <xs:extension base="md:string-Subtitle-Format">
      <xs:attribute name="SDImage" type="xs:boolean"/>
    </xs:extension>
  </xs:simpleContent>
</xs:complexType>

```

```

    <xs:attribute name="HDImage" type="xs:boolean"/>
  </xs:extension>
</xs:simpleContent>
</xs:complexType>
<xs:complexType name="DigitalAssetSubtitleData-type">
  <xs:sequence>
    <xs:element name="Format" type="md:DigitalAssetSubtitleFormat-type"
      minOccurs="0"/>
    <xs:element name="Description" type="xs:string" minOccurs="0"/>
    <xs:element name="Type" type="md:string-Subtitle-Type" maxOccurs="unbounded"/>
    <xs:element name="FormatType" type="md:string-Subtitle-FormatType"
      minOccurs="0"/>
    <xs:element name="Language" type="md:language-redefine"/>
    <xs:element name="Encoding" type="xs:anyType" minOccurs="0"/>
    <xs:element name="CardsetList" type="md:DigitalAssetCardsetList-type"
      minOccurs="0" maxOccurs="unbounded"/>
    <xs:element name="TrackReference" type="md:string-TrackReference"
      minOccurs="0"/>
    <xs:element name="TrackIdentifier" type="md:ContentIdentifier-type"
      minOccurs="0" maxOccurs="unbounded"/>
    <xs:element name="Private" type="md:PrivateData-type" minOccurs="0"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="DigitalAssetInteractiveEncoding-type">
  <xs:sequence>
    <xs:element name="RuntimeEnvironment" type="md:string-Interactive-Enc-
      RuntimeEnvironment"/>
    <xs:element name="FirstVersion" type="md:string-Interactive-Enc-Version"
      minOccurs="0"/>
    <xs:element name="LastVersion" type="md:string-Interactive-Enc-Version"
      minOccurs="0"/>
    <xs:any namespace="##other" minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>
<xs:group name="DigitalAssetInteractiveBase-group">
  <xs:sequence>
    <xs:element name="Type" type="md:string-Interactive-Type"/>
    <xs:element name="FormatType" type="md:string-Interactive-FormatType"
      minOccurs="0"/>
    <xs:element name="Language" type="md:language-redefine" minOccurs="0"/>
  </xs:sequence>
</xs:group>
<xs:complexType name="DigitalAssetInteractiveBaseData-type">
  <xs:sequence>
    <xs:group ref="md:DigitalAssetInteractiveBase-group"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="DigitalAssetInteractiveData-type">
  <xs:sequence>
    <xs:group ref="md:DigitalAssetInteractiveBase-group"/>
    <xs:element name="Encoding" type="md:DigitalAssetInteractiveEncoding-type"
      maxOccurs="unbounded"/>
    <xs:element name="TrackReference" type="md:string-TrackReference"
      minOccurs="0"/>
    <xs:element name="TrackIdentifier" type="md:ContentIdentifier-type"
      minOccurs="0" maxOccurs="unbounded"/>
    <xs:element name="Private" type="md:PrivateData-type" minOccurs="0"/>
  </xs:sequence>

```



```

</xs:sequence>
</xs:complexType>
<xs:complexType name="DigitalAssetSet-type">
  <xs:sequence>
    <xs:element name="Audio" type="md:DigitalAssetAudioData-type" minOccurs="0"
      maxOccurs="unbounded"/>
    <xs:element name="Video" type="md:DigitalAssetVideoData-type" minOccurs="0"
      maxOccurs="unbounded"/>
    <xs:element name="Subtitle" type="md:DigitalAssetSubtitleData-type"
      minOccurs="0" maxOccurs="unbounded"/>
    <xs:element name="Image" type="md:DigitalAssetImageData-type" minOccurs="0"
      maxOccurs="unbounded"/>
    <xs:element name="Interactive" type="md:DigitalAssetInteractiveData-type"
      minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="DigitalAssetMetadata-type">
  <xs:choice>
    <xs:element name="Audio" type="md:DigitalAssetAudioData-type"/>
    <xs:element name="Video" type="md:DigitalAssetVideoData-type"/>
    <xs:element name="Subtitle" type="md:DigitalAssetSubtitleData-type"/>
    <xs:element name="Image" type="md:DigitalAssetImageData-type"/>
    <xs:element name="Interactive" type="md:DigitalAssetInteractiveData-type"/>
  </xs:choice>
</xs:complexType>
<!-- *** Container description *** -->
<xs:complexType name="ContainerTrackMetadata-type">
  <xs:choice>
    <xs:element name="Audio" type="md:DigitalAssetAudioData-type"/>
    <xs:element name="Video" type="md:DigitalAssetVideoData-type"/>
    <xs:element name="Subtitle" type="md:DigitalAssetSubtitleData-type"/>
    <xs:element name="Image" type="md:DigitalAssetImageData-type"/>
    <xs:element name="Interactive" type="md:DigitalAssetInteractiveData-type"/>
    <xs:element name="Container" type="md:ContainerMetadata-type"/>
    <xs:element name="ExternalTrackReference"
      type="md:DigitalAssetExternalTrackReference-type"/>
    <xs:element name="InternalTrackReference" type="md:string-TrackReference"/>
  </xs:choice>
</xs:complexType>
<xs:simpleType name="ContainerProfile-type">
  <xs:restriction base="xs:string"/>
</xs:simpleType>
<xs:complexType name="ContainerSpecific-type">
  <xs:sequence>
    <xs:any namespace="##any" processContents="lax" minOccurs="0"
      maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>
<xs:simpleType name="ContainerType-type">
  <xs:restriction base="xs:string"/>
</xs:simpleType>
<xs:complexType name="ContainerMetadata-type">
  <xs:sequence>
    <xs:element name="Type" type="md:string-Container-Type" minOccurs="0"/>
    <xs:element name="Track" type="md:ContainerTrackMetadata-type"
      maxOccurs="unbounded"/>
    <xs:element name="Hash" type="md:Hash-type" minOccurs="0"

```

```

maxOccurs="unbounded"/>
<xs:element name="Size" type="xs:positiveInteger" minOccurs="0"/>
<xs:element name="ContainerSpecificMetadata" type="md:ContainerSpecific-type"
minOccurs="0"/>
<xs:any namespace="##other" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
</xs:sequence>
</xs:complexType>
<!-- *** Compilation Object *** -->
<xs:complexType name="CompObjClass-type">
  <xs:simpleContent>
    <xs:extension base="md:string-Compilation-CompilationClass">
      <xs:attribute name="hasOtherInclusions" type="xs:boolean" use="optional"/>
    </xs:extension>
  </xs:simpleContent>
</xs:complexType>
<xs:simpleType name="CompObjID-type">
  <xs:restriction base="md:id-type"/>
</xs:simpleType>
<xs:complexType name="CompObjEntry-type">
  <xs:sequence>
    <xs:element name="DisplayName" minOccurs="0" maxOccurs="unbounded">
      <xs:complexType>
        <xs:simpleContent>
          <xs:extension base="xs:string">
            <xs:attribute name="language" type="xs:language"/>
          </xs:extension>
        </xs:simpleContent>
      </xs:complexType>
    </xs:element>
    <xs:element name="EntryNumber" type="md:string-Compilation-EntryNumber"
minOccurs="0"/>
    <xs:element name="EntryClass" type="md:string-Compilation-EntryClass"
minOccurs="0"/>
    <xs:element name="Entry" type="md:CompObjEntry-type" minOccurs="0"
maxOccurs="unbounded"/>
    <xs:choice>
      <xs:element name="ContentID" type="md:ContentID-type"/>
      <xs:element name="BasicMetadata" type="md:BasicMetadata-type"/>
      <xs:any namespace="##other" processContents="lax"/>
    </xs:choice>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="CompObjData-type">
  <xs:complexContent>
    <xs:extension base="md:CompObj-type">
      <xs:sequence>
        <xs:element name="DisplayName" minOccurs="0" maxOccurs="unbounded">
          <xs:complexType>
            <xs:simpleContent>
              <xs:extension base="xs:string">
                <xs:attribute name="language" type="xs:language"/>
              </xs:extension>
            </xs:simpleContent>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

```

```

    <xs:attribute name="CompObjID" type="md:CompObjID-type"/>
  </xs:extension>
</xs:complexContent>
</xs:complexType>
</xs:complexType>
<xs:complexType name="CompObj-type">
  <xs:sequence>
    <xs:element name="Entry" type="md:CompObjEntry-type" maxOccurs="unbounded"/>
    <xs:element name="CompilationClass" type="md:CompObjClass-type" minOccurs="0"/>
  </xs:sequence>
</xs:complexType>
<!-- *** Prior Release Information *** -->
<xs:complexType name="ReleaseHistory-type">
  <xs:sequence>
    <xs:element name="ReleaseType">
      <xs:complexType>
        <xs:simpleContent>
          <xs:extension base="md:string-Release-ReleaseType">
            <xs:attribute name="wide" type="xs:boolean"/>
          </xs:extension>
        </xs:simpleContent>
      </xs:complexType>
    </xs:element>
    <xs:element name="DistrTerritory" type="md:Region-type" minOccurs="0"/>
    <xs:element name="Date">
      <xs:complexType>
        <xs:simpleContent>
          <xs:extension base="md:YearDateOrTime-type">
            <xs:attribute name="scheduled" type="xs:boolean"/>
          </xs:extension>
        </xs:simpleContent>
      </xs:complexType>
    </xs:element>
    <xs:element name="Description" type="xs:string" minOccurs="0"/>
    <xs:element name="ReleaseOrg" type="md:OrgName-type" minOccurs="0"
      maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>
<!-- -->
<!-- *** ADDITIONAL TYPES *** -->
<xs:simpleType name="Role-type">
  <xs:restriction base="xs:string"/>
</xs:simpleType>
<xs:complexType name="Money-type">
  <xs:simpleContent>
    <xs:extension base="xs:decimal">
      <xs:attribute name="currency" type="md:string-Money-currency"/>
    </xs:extension>
  </xs:simpleContent>
</xs:complexType>
<xs:complexType name="NVPairMoney-type">
  <xs:sequence>
    <xs:element name="Name" type="xs:string"/>
    <xs:element name="Value" type="md:Money-type"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="NVPair-type">
  <xs:sequence>

```

```

    <xs:element name="Name" type="xs:string"/>
    <xs:element name="Value" type="xs:string"/>
  </xs:sequence>
</xs:complexType>
<xs:simpleType name="ColorType-type">
  <xs:restriction base="xs:string">
    <xs:enumeration value="color"/>
    <xs:enumeration value="bandw"/>
    <xs:enumeration value="colorized"/>
    <xs:enumeration value="composite"/>
    <xs:enumeration value="unknown"/>
  </xs:restriction>
</xs:simpleType>
<xs:simpleType name="Region-string">
  <xs:restriction base="xs:string"/>
</xs:simpleType>
<xs:complexType name="Region-type">
  <xs:choice>
    <xs:element name="country">
      <xs:simpleType>
        <xs:restriction base="xs:string">
          <xs:pattern value="[A-Z][A-Z]"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:element>
    <xs:element name="countryRegion">
      <xs:simpleType>
        <xs:restriction base="xs:string">
          <xs:pattern value="[A-Z][A-Z]-[A-Z0-9]"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:element>
  </xs:choice>
</xs:complexType>
<xs:simpleType name="MadeForRegion-type">
  <xs:union memberTypes="md:MadeForRegionInternal-type md:Region-string"/>
</xs:simpleType>
<xs:simpleType name="MadeForRegionInternal-type">
  <xs:restriction base="xs:string">
    <xs:enumeration value="Domestic"/>
    <xs:enumeration value="International"/>
  </xs:restriction>
</xs:simpleType>
<xs:complexType name="OrgName-type">
  <xs:sequence>
    <xs:element name="DisplayName" type="xs:string"/>
    <xs:element name="SortName" type="xs:string" minOccurs="0"/>
    <xs:element name="AlternateName" type="xs:string" minOccurs="0"
      maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:attribute name="organizationID" type="md:orgID-type"/>
  <xs:attribute name="idType" type="xs:string"/>
</xs:complexType>
<xs:complexType name="PersonName-type">
  <xs:sequence>
    <xs:element name="DisplayName" maxOccurs="unbounded">
      <xs:complexType>

```

```

        <xs:simpleContent>
          <xs:extension base="xs:string">
            <xs:attribute name="language" type="xs:language"/>
          </xs:extension>
        </xs:simpleContent>
      </xs:complexType>
    </xs:element>
    <xs:element name="SortName" minOccurs="0" maxOccurs="unbounded">
      <xs:complexType>
        <xs:simpleContent>
          <xs:extension base="xs:string">
            <xs:attribute name="language" type="xs:language"/>
          </xs:extension>
        </xs:simpleContent>
      </xs:complexType>
    </xs:element>
    <xs:element name="FirstGivenName" type="xs:string" minOccurs="0"/>
    <xs:element name="SecondGivenName" type="xs:string" minOccurs="0"/>
    <xs:element name="FamilyName" type="xs:string" minOccurs="0"/>
    <xs:element name="Suffix" type="md:string-Name_Suffix" minOccurs="0"/>
    <xs:element name="Moniker" type="xs:string" minOccurs="0"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="ContactInfo-type">
  <xs:sequence>
    <xs:element name="Name" type="xs:string"/>
    <xs:element name="PrimaryEmail" type="xs:string"/>
    <xs:element name="AlternateEmail" type="xs:string" minOccurs="0"
maxOccurs="unbounded"/>
    <xs:element name="Address" type="xs:string" minOccurs="0"
maxOccurs="unbounded"/>
    <xs:element name="Phone" minOccurs="0" maxOccurs="unbounded">
      <xs:complexType>
        <xs:simpleContent>
          <xs:extension base="xs:string">
            <xs:attribute name="type" type="xs:string"/>
          </xs:extension>
        </xs:simpleContent>
      </xs:complexType>
    </xs:element>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="PersonIdentifier-type">
  <xs:sequence>
    <xs:element name="Identifier" type="xs:string"/>
    <xs:element name="Namespace" type="md:string-Idenifier_Namespace"/>
    <xs:element name="ReferenceLocation" type="xs:anyURI" minOccurs="0"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="DateTimeRange-type">
  <xs:sequence>
    <xs:element name="Start" type="xs:dateTime"/>
    <xs:element name="End" type="xs:dateTime"/>
  </xs:sequence>
</xs:complexType>
<xs:simpleType name="YearDateOrTime-type">
  <xs:union memberTypes="xs:gYear xs:date xs:dateTime"/>

```

```

</xs:simpleType>
<xs:complexType name="Hash-type">
  <xs:simpleContent>
    <xs:extension base="md:string-Hash">
      <xs:attribute name="method" type="md:string-Hash-method"/>
    </xs:extension>
  </xs:simpleContent>
</xs:complexType>
<xs:complexType name="CompanyCredits-type">
  <xs:sequence>
    <xs:element name="DisplayString" maxOccurs="unbounded">
      <xs:complexType>
        <xs:simpleContent>
          <xs:extension base="xs:string">
            <xs:attribute name="language" type="xs:language"/>
          </xs:extension>
        </xs:simpleContent>
      </xs:complexType>
    </xs:element>
    <xs:element name="Region" type="md:Region-type" minOccurs="0"
      maxOccurs="unbounded"/>
    <xs:element name="DisplaySequence" type="xs:integer" minOccurs="0"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="GroupingEntity-type">
  <xs:sequence>
    <xs:element name="Type" type="xs:string"/>
    <xs:element name="GroupIdentity" type="xs:string"/>
    <xs:element name="DisplayName" maxOccurs="unbounded">
      <xs:complexType>
        <xs:simpleContent>
          <xs:extension base="xs:string">
            <xs:attribute name="language" type="xs:language"/>
          </xs:extension>
        </xs:simpleContent>
      </xs:complexType>
    </xs:element>
    <xs:element name="Region" type="md:Region-type" minOccurs="0"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="PrivateData-type">
  <xs:sequence>
    <xs:any namespace="##any" processContents="lax" minOccurs="0"
      maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>
<!-- -->
<!-- *** REDEFINE TYPES *** -->
<!-- -->
<!-- *** General Redfine Types *** -->
<xs:simpleType name="language-redefine">
  <xs:restriction base="xs:language"/>
</xs:simpleType>
<!-- *** Identifier Redefine types *** -->
<xs:simpleType name="string-ContentID-Namespace">
  <xs:restriction base="xs:string"/>
</xs:simpleType>

```

```
<xs:simpleType name="string-ContentID-Identifier">
  <xs:restriction base="xs:string"/>
</xs:simpleType>
<!-- *** Basic Metadata Redefine types *** -->
<xs:simpleType name="string-Genre">
  <xs:restriction base="xs:string"/>
</xs:simpleType>
<xs:simpleType name="string-Genre_id">
  <xs:restriction base="xs:string"/>
</xs:simpleType>
<xs:simpleType name="string-ArtReference_resolution">
  <xs:restriction base="xs:string"/>
</xs:simpleType>
<xs:simpleType name="string-DisplayIndicators">
  <xs:restriction base="xs:string">
    <xs:enumeration value="CC"/>
    <xs:enumeration value="F"/>
    <xs:enumeration value="P"/>
    <xs:enumeration value="DD"/>
    <xs:enumeration value="SAP"/>
    <xs:enumeration value="DVS"/>
  </xs:restriction>
</xs:simpleType>
<xs:simpleType name="string-Keyword">
  <xs:restriction base="xs:string"/>
</xs:simpleType>
<xs:simpleType name="string-JobFunction">
  <xs:restriction base="xs:string"/>
</xs:simpleType>
<xs:simpleType name="string-JobFunction_scheme">
  <xs:restriction base="xs:string"/>
</xs:simpleType>
<xs:simpleType name="string-JobDisplay">
  <xs:restriction base="xs:string"/>
</xs:simpleType>
<xs:simpleType name="string-Name_Suffix">
  <xs:restriction base="xs:string"/>
</xs:simpleType>
<xs:simpleType name="string-Idenifier_Namespace">
  <xs:restriction base="xs:string"/>
</xs:simpleType>
<xs:simpleType name="string-TitleAlternate_type">
  <xs:restriction base="xs:string"/>
</xs:simpleType>
<xs:simpleType name="string-WorkType">
  <xs:restriction base="xs:string"/>
</xs:simpleType>
<xs:simpleType name="string-WorkTypeDetail">
  <xs:restriction base="xs:string"/>
</xs:simpleType>
<xs:simpleType name="string-PictureFormat">
  <xs:restriction base="xs:string"/>
</xs:simpleType>
<xs:simpleType name="string-AspectRatio">
  <xs:restriction base="xs:string"/>
</xs:simpleType>
<xs:simpleType name="string-AssociatedOrg_role">
```



```
<xs:restriction base="xs:string"/>
</xs:simpleType>
<!-- *** Digital Asset Redefine types *** -->
<xs:simpleType name="string-TrackReference">
  <xs:restriction base="xs:string"/>
</xs:simpleType>
<xs:simpleType name="string-Audio-Type">
  <xs:restriction base="xs:string"/>
</xs:simpleType>
<xs:simpleType name="string-Audio-Channels">
  <xs:restriction base="xs:string"/>
</xs:simpleType>
<xs:simpleType name="string-Audio-Enc-Codec">
  <xs:restriction base="xs:string"/>
</xs:simpleType>
<xs:simpleType name="string-Audio-Enc-CodecType">
  <xs:restriction base="xs:string"/>
</xs:simpleType>
<xs:simpleType name="string-Audio-Enc-VBR">
  <xs:restriction base="xs:string"/>
</xs:simpleType>
<xs:simpleType name="string-Audio-Enc-ChannelMapping">
  <xs:restriction base="xs:string"/>
</xs:simpleType>
<xs:simpleType name="string-Video-Type">
  <xs:restriction base="xs:string"/>
</xs:simpleType>
<xs:simpleType name="string-Video-PictureFormat">
  <xs:restriction base="xs:string"/>
</xs:simpleType>
<xs:simpleType name="string-Video-Enc-Codec">
  <xs:restriction base="xs:string"/>
</xs:simpleType>
<xs:simpleType name="string-Video-Enc-CodecType">
  <xs:restriction base="xs:string"/>
</xs:simpleType>
<xs:simpleType name="string-Video-Enc-MProfile">
  <xs:restriction base="xs:string"/>
</xs:simpleType>
<xs:simpleType name="string-Video-Enc-MLevel">
  <xs:restriction base="xs:string"/>
</xs:simpleType>
<xs:simpleType name="string-Video-Enc-VBR">
  <xs:restriction base="xs:string"/>
</xs:simpleType>
<xs:simpleType name="string-Video-Pic-AspectRatio">
  <xs:restriction base="xs:string"/>
</xs:simpleType>
<xs:simpleType name="string-Video-Pic-PixelAspect">
  <xs:restriction base="xs:string">
    <xs:enumeration value="NTSC"/>
    <xs:enumeration value="PAL"/>
    <xs:enumeration value="square"/>
    <xs:enumeration value="other"/>
  </xs:restriction>
</xs:simpleType>
<xs:simpleType name="string-Video-Pic-ColorSampling">
```



```
<xs:restriction base="xs:string"/>
</xs:simpleType>
<xs:simpleType name="string-Video-Pic-Colorimetry">
  <xs:restriction base="xs:string"/>
</xs:simpleType>
<xs:simpleType name="string-Video-Pic-FrameRate-multiplier">
  <xs:restriction base="xs:string">
    <xs:enumeration value="1000/1001"/>
  </xs:restriction>
</xs:simpleType>
<xs:simpleType name="string-Video-Pic-FrameRate-timecode">
  <xs:restriction base="xs:string"/>
</xs:simpleType>
<xs:simpleType name="string-Video-Pic-Progressive-scanOrder">
  <xs:restriction base="xs:string">
    <xs:enumeration value="BFF"/>
    <xs:enumeration value="TFF"/>
    <xs:enumeration value="PPF"/>
  </xs:restriction>
</xs:simpleType>
<xs:simpleType name="string-Video-Pic-Type3D">
  <xs:restriction base="xs:string"/>
</xs:simpleType>
<xs:simpleType name="string-Video-SubLang-type">
  <xs:restriction base="xs:string"/>
</xs:simpleType>
<xs:simpleType name="string-Subtitle-Format">
  <xs:restriction base="xs:string"/>
</xs:simpleType>
<xs:simpleType name="string-Subtitle-Type">
  <xs:restriction base="xs:string"/>
</xs:simpleType>
<xs:simpleType name="string-Subtitle-FormatType">
  <xs:restriction base="xs:string"/>
</xs:simpleType>
<xs:simpleType name="string-Subtitle-Enc-Codec">
  <xs:restriction base="xs:string"/>
</xs:simpleType>
<xs:simpleType name="string-Subtitle-Enc-CodecType">
  <xs:restriction base="xs:string"/>
</xs:simpleType>
<xs:simpleType name="string-Image-Encoding">
  <xs:restriction base="xs:string"/>
</xs:simpleType>
<xs:simpleType name="string-Interactive-Type">
  <xs:restriction base="xs:string"/>
</xs:simpleType>
<xs:simpleType name="string-Interactive-FormatType">
  <xs:restriction base="xs:string"/>
</xs:simpleType>
<xs:simpleType name="string-Interactive-Enc-RuntimeEnvironment">
  <xs:restriction base="xs:string"/>
</xs:simpleType>
<xs:simpleType name="string-Interactive-Enc-Version">
  <xs:restriction base="xs:string"/>
</xs:simpleType>
<xs:simpleType name="string-CardsetList-Type">
```

```
<xs:restriction base="xs:string"/>
</xs:simpleType>
<xs:simpleType name="string-CardsetList-Location">
  <xs:restriction base="xs:string"/>
</xs:simpleType>
<xs:simpleType name="string-Cardset-Type">
  <xs:restriction base="xs:string"/>
</xs:simpleType>
<xs:simpleType name="string-Watermark_Vendor">
  <xs:restriction base="xs:string"/>
</xs:simpleType>
<!-- *** Other *** -->
<xs:simpleType name="string-NotRated-condition">
  <xs:restriction base="xs:string"/>
</xs:simpleType>
<xs:simpleType name="string-Rating-System">
  <xs:restriction base="xs:string"/>
</xs:simpleType>
<xs:simpleType name="string-Release-ReleaseType">
  <xs:restriction base="xs:string"/>
</xs:simpleType>
<xs:simpleType name="string-Compilation-CompilationClass">
  <xs:restriction base="xs:string"/>
</xs:simpleType>
<xs:simpleType name="string-Compilation-EntryNumber">
  <xs:restriction base="xs:string"/>
</xs:simpleType>
<xs:simpleType name="string-Compilation-EntryClass">
  <xs:restriction base="xs:string"/>
</xs:simpleType>
<xs:simpleType name="string-Money-currency">
  <xs:restriction base="xs:string"/>
</xs:simpleType>
<xs:simpleType name="string-Container-Type">
  <xs:restriction base="xs:string"/>
</xs:simpleType>
<xs:simpleType name="string-Hash-method">
  <xs:restriction base="xs:string"/>
</xs:simpleType>
<xs:simpleType name="string-Hash">
  <xs:restriction base="xs:string"/>
</xs:simpleType>
</xs:schema>
```



entertainment merchants association

16530 Ventura Blvd., Suite 400

Encino, CA 91436

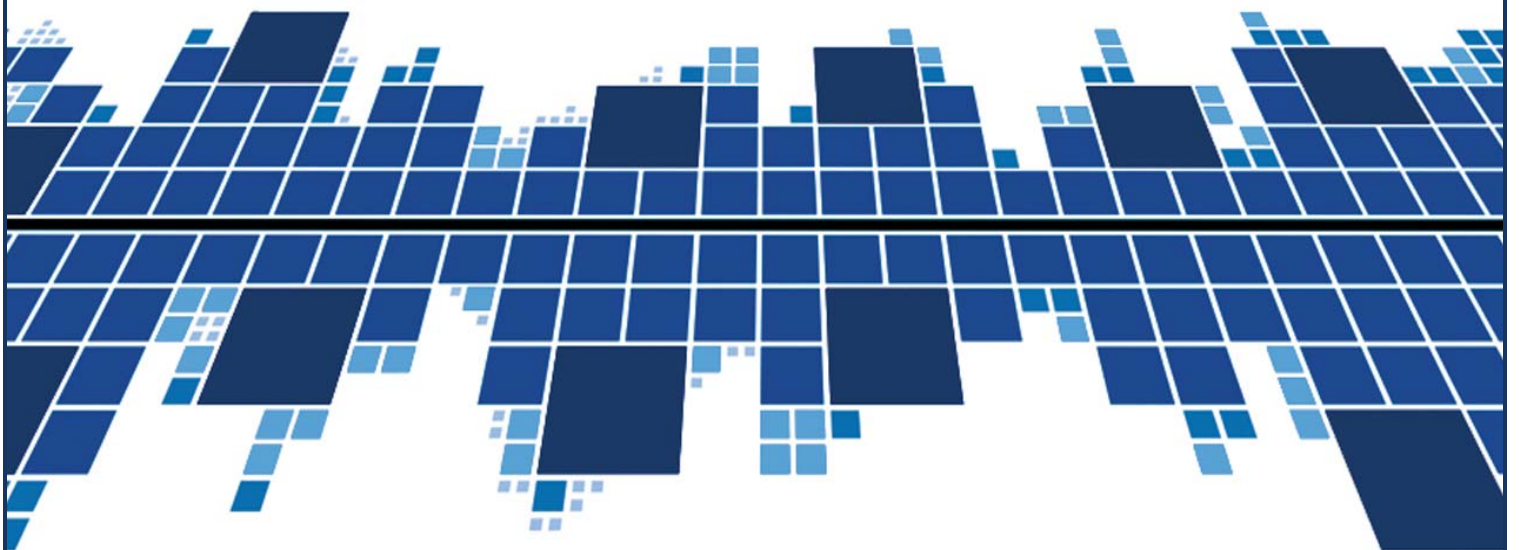
818.385.1500

[www.entmerch.org](http://www.entmerch.org)

**Best Practices & Specifications  
For the Delivery of  
Mezzanine Files  
for  
Digital Audio-Visual Distribution**

DRAFT

January 8, 2013





To All Stakeholders in the Digital Delivery of Audio Visual Entertainment –

EMA's Digital Supply Chain Committee was formed to seek solutions to optimize the delivery of digital content and its associated components from the content providers through service providers and then through retailers/distributors ultimately to the consumer. Opportunities to improve and standardize the consumer experience will be explored, as will new technology to bring speed and efficiencies to this supply chain.

One of the initial opportunities identified was to draft a recommended standard mezzanine file specification.

Why?

- At the request of their customers, content providers and post-house are creating mezzanine files unique to each of their retail partners. This causes unnecessary costs in the supply chain and constrains the flow of new content.
- There is a demand to make more content available for digital distribution more quickly. Sales are lost if content isn't available to be merchandised.
- Today's ecosystem is too manual. Standardization will facilitate automation, reducing costs and increasing speed.
- Quality control issues slow down today's processes. Creating one standard mezzanine file instead of many files for the same content should reduce the quantity of errors. And, when an error does occur and is caught by a single customer, it can be corrected for all retailers/distributors.

This EMA file spec has been developed over a series of meetings among retailers/distributors addressing their needs to fulfill delivery to the consumer. It is in its final draft phase, and stakeholders are requested to provide feedback and input before a final recommended specification is adopted.

Mark Fisher, EVP, EMA, on behalf of EMA's Digital Supply Chain Committee



# EMA MEZZANINE FILE CREATION SPECIFICATION and BEST PRACTICES

## Version Beta 1

For encoding of contribution quality content delivered to EMA member companies.

Developed by the Digital Supply Chain Committee (Mezzanine File Work Group)  
of the Entertainment Merchants Association.

[introduction](#)

[technical goals](#)

[content preparation considerations](#)

[Video preparation](#)

[timed-text](#)

[acceptable formatting](#)

[Video Encoding](#)

[H.264 video encoding all resolutions](#)

[TODO for steven](#)

[h.264 high definition specific requirements](#)

[h.264 standard definition specific requirements](#)

[mpeg-2 video encoding all resolutions](#)

[mpeg-2 high definition specific requirements](#)

[mpeg-2 standard definition specific requirements](#)

[Sample Aspect Ratio](#)

[Native to the source content](#)

[Subsampling](#)

[4:2:2 required](#)

[Audio Encoding](#)

[Container](#)

[MPEG-2 File Format Specifications](#)

[Helpful Hints](#)

## introduction

- This is a spec that industry veterans from post production, delivery platforms, software vendors, and content creator's technical operations departments have come to agreement on.
- It is intended to create a common baseline format that all agreed upon parties can create and ingest.
- It is not intended to be academically the "best" spec for quality, but instead, for practicality and compatibility with sufficient quality.
- EMA vendor companies have agreed that they are able to receive these types of files. Parties involved may however, want to set up on a case by case basis special alternate formats and workflows with post production or content owners for various reasons.

## technical goals

- Lower bitrates than MPEG-2 or ProRes based deliverables for practical storage, transmission, and manipulation of these assets.
- Support for uncompressed or compressed audio.
- Reasonable encoding and decoding speed
- No or low cost to begin creation or acceptance of assets in this spec.
- Requires less downstream processing and storage
- "Creative" decisions are made by content owners or post production stakeholders rather than resellers.

## status

- This spec is at the stage of a public draft / beta phase
- File creation, exchange, and ingestion can begin.
- Feedback to the EMA working group.

## content preparation considerations

Because one of the goals of this specification is to improve output quality while increasing consistency of results, it is important to limit the number of allowable standards for content formatting and have common expectations for preprocessing

### Video preparation



It is always preferable to receive content in its native format, meaning the format in which it was initially produced and not necessarily the format in which it was distributed. For example, if a TV program was shot on film at 23.976p, but then was initially broadcast in 59.94p, it would be desirable to receive this asset in its native 23.976p progressive frame rate.

Content must end and begin with at least ½ second of black with silence, and no more of 1 second of black with silence. No head or tail content like slates, bars and tone, etcetera are to be included with the mezzanine; the content should be only what the viewer would see and hear. The content must not contain more than 2-seconds of commercial blacks within program. Also, content must not contain any advertisements, commercials, VITC (Vertical Interval TimeCode), horizontal or vertical blanking lines.

Likewise, it is also desirable to have any mattes (letterbox or pillarbox), wide horizontal blanking, VBI, or other empty edge space to be removed from picture so that the mezzanine contains only active picture area. This cropping requirement means an HD 1920x1080 film produced with a 2.35 matte would require cropping to a 1920x816 resolution.

These cropping operations may result in delivering content with non-standard resolutions such as 638x474 as opposed to 640x480. This is why all content (HD) will be encoded using square pixels. Therefore, no misinterpretation of pixel aspect ratio can occur which might lead to an incorrect presentation of the display aspect ratio. All resolutions should be mod 4.

For titles that have burned-in subtitles, the subtitles must not be in the matted area and must be within the active pixel region of the content.

Full screen SD content that is interlaced should be encoded using a pixel aspect ratio native to the source material and not cropped. Therefore, for example, NTSC content in full screen 4.3 or widescreen NTSC content in 16.9 would both be encoded in 720x480 (or 720x486, 720x576 etc).

Progressive SD content should be cropped to its native active pixel resolution, removing any letterboxing or pillarboxing. The pixels must be square.

Mixed mode content should be conformed to a single progressive mode. For example a 24p SD TV show with a 29.97i intro, the whole file should become a 24p. For a special feature clip that is made up primarily of 29.97i behind-the-scenes content, with 24p clips occasionally, the file should become 29.97i. Content that is originally mixed mode, but has been conformed to progressive, must be identified in the associated EMA file as "TODO: Andy defines how he wants mixed mode content identified in metadata." 2:3 pulldown flags should not be used.

Beyond frame rates and mattes, there are an enormous range of production and post-production artifacts that need to be considered. Future versions of this document will attempt to make more formal recommendations regarding handling challenging content. We have included early recommendations below.

## timed-text

Subtitles and other timed text events will be addressed in the "[CLOSED CAPTIONING OF IP-DELIVERED VIDEO PROGRAMMING BEST PRACTICES](#)" document.

## acceptable formatting

#	video codec	frame rate	scan type	notes
1	AVC / H.264	23.976	Progressive	Preferred format for all HD and SD resolutions if content was natively produced in 23.976p.
2	AVC / H.264	59.94	Progressive	Only if sourced from true 59.94p content such as sports networks. No repeated frames for telecine or up/cross-conversions from 29.97i.
3	AVC / H.264	29.97	Progressive	For content that is native 29.97 progressive. Or content that is true 29.97 interlaced and has been properly deinterlaced to 29.97 progressive. No inter-field motion and no repeated frames.
4	MPEG-2	23.976	Progressive	A fallback option. ONLY allowed if vendor cannot provide suitable AVC encoding.
5	MPEG-2	59.94	Progressive	Same restrictions as #2 above. ONLY allowed if vendor cannot provide suitable AVC encoding.
6	MPEG-2	29.97	Progressive	Same restrictions as #3 above. ONLY allowed if vendor cannot provide suitable AVC encoding.

7	AVC / H.264	29.97	Interlaced	Only for content that was natively interlaced or converted to interlaced and no suitable recovery method is available to inverse telecine or deinterlace.
8	MPEG-2	29.97	Interlaced	Same restrictions as #7 above. ONLY allowed if vendor cannot provide suitable AVC encoding.

## Video Encoding

### H.264 video encoding all resolutions

Codec	H.264 / AVC / MPEG-4 Part 10
Bit-depth	8-bit
Subsampling / color space	4:2:0
Frame rate:	To be determined by source content and any pre-processing. Fixed frame rate only (aka constant frame rate), no variable frame rate.
Scan type:	Progressive. If interlaced material is delivered for Format #7, it must have the Interlace Mode correctly set (both MBAFF and Interlaced are acceptable) and the Field Order must be correctly assigned (Top Field First or Bottom Field First) matching the encoded content.
Resolution	To be determined by cropping performed on matted content
Sample Aspect Ratio (pixel aspect ratio)	Pixel Aspect Ratio
Resolution	Maximum of 1920 pixels wide and 1080 pixels tall for HD
Rate control	Constant Quantizer (QP) or Constant Rate Factor (CRF) or Two Pass Variable Bitrate (VBR) Resulting bitrate must stay within the constraints of the corresponding H.264 Profile/Level.

Entropy encoding	CAVLC
Scene change detection	Enabled (Variable GOP)
GOP length	2 seconds max, with Scene Change Detection enabled.
GOP style	Closed
B Frames	Maximum of 3 consecutive B Frames
Reference frames	Maximum of 3 reference frames
Reference B frames	If used, must adhere to strict pyramid structure
Required stream flags	aspect_ratio_idc, field_flag, sequence_parameter_set, picture_parameter_set

#### h.264 high definition specific requirements

H.264 Profile	High
H.264 Levels	4.1 for standard HD content 5.2 for 2k or 4k content
Bitrate Constraint	62,500 kbps maximum (variable bitrate) 240,000 kbps maximum for 2k or 4k content
Resolution Constraint	At least 1280 wide <b>OR</b> 720 tall to be considered HD widescreen content. At least 960x720 to be considered HD for 4:3 content. Resolutions more than 1024 wide <b>OR</b> 576 tall should use the HD spec.
Pixel Aspect Ratio	HD must be 1:1 (aka square pixels) only. No anamorphic squeezing allowed (e.g. 16:9 1440x1080).
Required stream flags	aspect_ratio_idc, field_flag, sequence_parameter_set, picture_parameter_set

#### h.264 standard definition specific requirements

H.264 Profile	High
H.264 Level	3.2
Bitrate constraint	25,000 kbps maximum

Resolution constraint	Maximum 1024x576 (square pixel 16:9 PAL) Typical NTSC: 720x480 or 704x480 (486 lines not supported) Typical PAL: 720x576 or 704x576 at 62.5 fps maximum (full resolution square pixel SD PAL)
Pixel aspect ratio	NTSC/PAL pixel aspect ratios are allowed for 720 and 704 wide content to match the source.
Required stream flags	aspect_ratio_idc, field_flag, sequence_parameter_set, picture_parameter_set

### mpeg-2 video encoding all resolutions

Codec	MPEG-2
Frame rate	To be determined by source content and any pre-processing.
Scan type	Progressive.  If interlaced material is delivered for Format #8, it must have the Picture Structure correctly set to Interlaced and the Field Order must be correctly assigned (Top Field First or Bottom Field First) matching the encoded content.
Resolution	To be determined by cropping performed on matted content.
GOP Length	Maximum of 0.6 seconds Scene Change Detection allowed and recommended
GOP style	Closed for all GOPs
B Frames	Maximum of 2 consecutive B Frames (a P-Frame distance of 3)

### mpeg-2 high definition specific requirements

MPEG-2 Profile	Main or High
MPEG-2 Level	High
Avg. Bitrate	50,000 to 80,000 kbps

Sample Aspect Ratio	Square pixels
Color space	4:2:2 preferred 4:2:0 accepted.

### mpeg-2 standard definition specific requirements

MPEG-2 Profile	Main or High
MPEG-2 Level	Main
Avg. Bitrate	10,000 kbps maximum
Peak Bitrate	15,000 kbps maximum
Sample Aspect Ratio	Native to the source content
Subsampling	4:2:2 required

### Audio Encoding

Codec	PCM or MPEG-4 AAC-LC								
Sample Rate	48000 Hz								
Endianness	Little endian								
Bits per sample	16 or 24 bit, following the source.								
Channel configuration	<p>Mono</p> <table border="1" style="margin-left: 40px;"> <tr> <td>1</td> <td>Mono</td> </tr> <tr> <td>2</td> <td>Mono</td> </tr> </table> <p>Stereo</p> <table border="1" style="margin-left: 40px;"> <tr> <td>1</td> <td>Left</td> </tr> <tr> <td>2</td> <td>Right</td> </tr> </table>	1	Mono	2	Mono	1	Left	2	Right
1	Mono								
2	Mono								
1	Left								
2	Right								

	<p>5.1 Surround</p> <table border="1" data-bbox="686 260 982 648"> <tr><td>1</td><td>Left Front</td></tr> <tr><td>2</td><td>Right Front</td></tr> <tr><td>3</td><td>Center</td></tr> <tr><td>4</td><td>LFE</td></tr> <tr><td>5</td><td>Left Surround</td></tr> <tr><td>6</td><td>Right Surround</td></tr> </table> <p>7.1 surround</p> <table border="1" data-bbox="686 716 1057 1234"> <tr><td>1</td><td>Left Front</td></tr> <tr><td>2</td><td>Right Front</td></tr> <tr><td>3</td><td>Center</td></tr> <tr><td>4</td><td>LFE</td></tr> <tr><td>5</td><td>Left Surround</td></tr> <tr><td>6</td><td>Right Surround</td></tr> <tr><td>7</td><td>Left Back Surround</td></tr> <tr><td>8</td><td>Right Back Surround</td></tr> </table>	1	Left Front	2	Right Front	3	Center	4	LFE	5	Left Surround	6	Right Surround	1	Left Front	2	Right Front	3	Center	4	LFE	5	Left Surround	6	Right Surround	7	Left Back Surround	8	Right Back Surround
1	Left Front																												
2	Right Front																												
3	Center																												
4	LFE																												
5	Left Surround																												
6	Right Surround																												
1	Left Front																												
2	Right Front																												
3	Center																												
4	LFE																												
5	Left Surround																												
6	Right Surround																												
7	Left Back Surround																												
8	Right Back Surround																												
Minimum AAC Bitrates	Mono: 128 Stereo: 192 5.1: 384																												

## Container

Container type	<p>(see specification at <a href="https://developer.apple.com/library/mac/#documentation/QuickTime/QTFF/QTFFChap3/qtff3.html#//apple_ref/doc/uid/TP40000939-CH205-SW1">https://developer.apple.com/library/mac/#documentation/QuickTime/QTFF/QTFFChap3/qtff3.html#//apple_ref/doc/uid/TP40000939-CH205-SW1</a>)</p> <p>Video (seen in spec table 4-1): Only allowed video sample description atom type must be 'avc1'</p> <p>Required:</p>
----------------	--

	<p>'pasp' 'fiel'</p> <p>Audio types allowed(seen in spec table 4-7): in24 lpcm twos sowt</p> <p>Other allowed atom types: 'tmcd'</p> <p>Not allowed: 'gama' 'clap' 'sprt' 'twen' 'load' 'hint' 'elst' 'qd3d' 'strm' 'stds' 'rtp ' 'vrsc' 'imgp' 'impn' 'vrni' Self-contained .mov</p>						
Container extension	.mov						
Metadata atom location	Located at the beginning of the file. (enable "Fast Start"if available")						
Stream order	<table border="1"> <tr> <td>Stream 0</td> <td>AVC video</td> </tr> <tr> <td>Stream 1</td> <td>PCM audio stereo</td> </tr> <tr> <td>Stream 2</td> <td>PCM audio surround</td> </tr> </table>	Stream 0	AVC video	Stream 1	PCM audio stereo	Stream 2	PCM audio surround
Stream 0	AVC video						
Stream 1	PCM audio stereo						
Stream 2	PCM audio surround						
Language labels	ISO 639-2						

## MPEG-2 File Format Specifications



This section covers Fallback Formats #4, 5, 6, and 8 in Table 2 above. These formats are only to be used if suitable results cannot be achieved by a vendor for delivering Preferred Formats #1-3.

## Helpful Hints

- Do not assume that your encoder and multiplexing software is generating files with all of the required metadata properly populated. Double check this explicitly.
- Do not assume that the metadata in all containers will be the same. Files with progressive video streams require different container data than interlaced streams.
- Do not assume that this is your father's web player. Internet capabilities constantly improve. Use the best available elements and don't pre-filter.

## sample materials

- Free high quality source footage for testing
  - Tears of Steel, available [here](#)
    - high quality 1920 x 1080 @ 24p source material
    - lossless sources available for audio and video
    - stereo and surround tracks available
- Sample file creating using [libav](#) and [x264](#)
  - Take the main video track out of the webm version of tears of steel, and turn it into an uncompressed file.
    - `~/libav/avconv -i tears_of_steel_1080p.webm -f yuv4mpepipe -pix_fmt yuv420p -vf scale tears_of_steel_uncompressed_1920x800-master.y4m`
  - Take the uncompressed file and compress it using x264
    - `~/x264/x264 tears_of_steel_uncompressed_1920x800-master.y4m --crf 8 --profile high --b-pyramid strict --bframes 3 --preset medium --level 41 --ref 3 --no-cabac --vbv-maxrate 62500 --vbv-bufsize 125000 --trellis 2 -o x264tears.mkv`
  - Encode audio, and mux it together with the h.264 video stream, and create a final EMA spec file, including metadata.
    - `avconv -i x264tears.mkv -i tearsofsteel-stereo.flac -i tearsofsteel-surround.flac -c:v:0 copy -c:a:0 pcm_s24le -c:a:1 pcm_s24le -map 0:0 -map 1:0 -map 2:0 -t 60 -metadata title="Tears of Steel" -metadata artist="Blender Foundation" -metadata date="2012" -metadata copyright="(CC) Blender Foundation | mango.blender.org" -metadata license="http://creativecommons.org/licenses/by/3.0/" -metadata description="EMA Mezzanine Encoding Spec Example File Beta 1" -metadata:s:a:0 language=eng -metadata:s:a:0 description="DVD stereo audio`

```
mix" -metadata:s:a:1 language=eng -metadata:s:a:1 description="DVD surround  
5.1 mix" tearsmuxedoutputtest.mov
```

- Move the atom to the front of the file using [qtfastsart](#)
  - `~/qtfaststart/bin/qtfaststart tears-of-steel-ema-example-file.mov qtfaststar-tears-of-steel-ema-example-file.mov`

DRAFT



entertainment merchants association

16530 Ventura Blvd., Suite 400

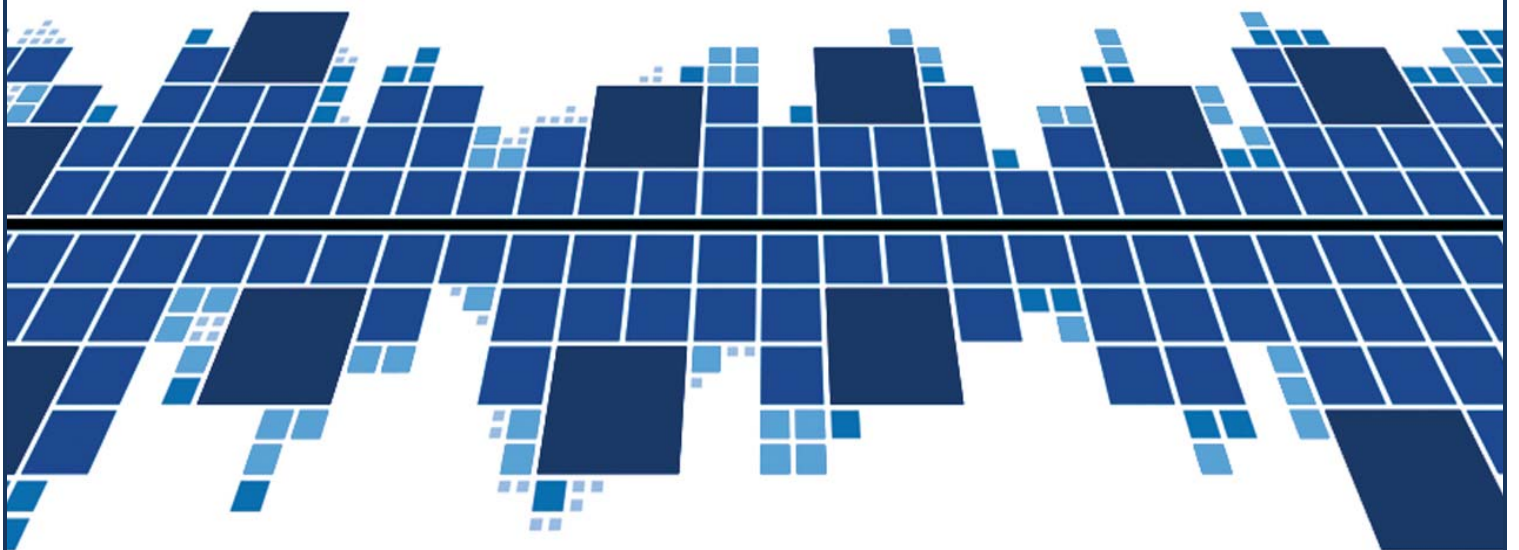
Encino, CA 91436

818.385.1500

[www.entmerch.org](http://www.entmerch.org)

**Best Practices & Specifications  
For the Delivery of  
Image Artwork  
for  
Digital Audio-Visual Distribution**

January 8, 2013







## **INTRODUCTION**

EMA's Digital Supply Chain Committee was formed to seek solutions to optimize the delivery of digital content and its associated components from the content providers through service providers and then through retailers/distributors ultimately to the consumer. Opportunities to improve and standardize the consumer experience will be explored, as will new technology to bring speed and efficiencies to this supply chain.

One of the initial opportunities identified was to draft best practices for, and a recommended standard set of, image artwork files for digital audiovisual content.

### **Why is it needed?**

- At the request of their customers, content providers are creating image files unique to each of their retail partners. This causes unnecessary costs in the supply chain and often delays the availability of new products to the consumer.
- Often times, individual retailers/distributors will edit and adapt studio-supplied image files to meet their merchandising needs. This is usually followed by a lengthy manual chain of re-edits and approvals for each piece of artwork, wasting a lot of staff time and often delaying the proper merchandising of content.
- Today's ecosystem is too manual. Standardization will facilitate automation, reducing costs and increasing speed.
- Quality control issues slow down today's processes. Creating a standard set of image files instead of many alternative files should reduce the quantity of errors. And, when an error does occur and is caught by a single customer, it can be corrected for all retailers/distributors.

This EMA file spec has been developed over a series of meetings among retailers/distributors addressing their needs to fulfill delivery to the consumer. It is anticipated that all participating retailers/distributors will be able to fulfill their normal needs within this set of image files.



# EMA's Best Practices & Specs for the Delivery of Image Artwork for Digital Audio-Visual Distribution

## Movies

### Sizes Required

Box Art - 3:4 - Minimum of 1000 pixels wide

Horizontal Image - 4:3 and 16:9 - Minimum of 2000 pixels wide

### File Format

Both a layered PSD file and either a png or jpeg need to be available.

PSD files should contain any titles, branding, or text in a separate layer. (Key art with title treatment layers removed should not show major gaps in design.)

### Color

RGB

TV Safe Colors

### Image Guidelines

No Tag lines

No Ratings

No Critic Quotes

No Dates

No Cast/Credits

Image must be recognizable from 10ft away on an SD TV that is 38-40 inches

If key artwork alternative, image provided must have one focal point (i.e., not too crowded/busy - 1 - 2 people max.)

### Naming

Use EIDR if available

If no EIDR available, please refer to the naming conventions agreed upon by the Studio/Network and Digital Distributor

### Foreign Releases

Need the localized name

Would prefer art used in the local territories

### No key art available

Choose an image to be consistent across platforms such as a still frame or images from other promotional materials

### Additional Images

Minimum of two stills in each of above sizes that are relevant to the audience (i.e., violent vs. conservative image.)

## **TV Shows**

### **Sizes Required**

Box Art - 3:4 - Minimum of 1000 pixels wide

Horizontal Image - 4:3 or 16:9 - Minimum of 2000 pixels wide

Square Image - 1:1- 800x800 pixels

### **File Format**

Both a layered PSD file and either a png or jpeg need to be available.

PSD files should contain any titles, branding, or text in a separate layer. (Key art with title treatment layers removed should not show major gaps in design.)

### **Series/Season/Episode Requirements**

Required: Series and Season artwork

Option: Episode artwork

### **Color**

RGB

TV Safe Colors

### **Image Guidelines**

No Tag lines

No Ratings

No Critic Quotes

No Dates

No Cast/Credits

Image must be recognizable from 10ft away on an SD TV that is 38-40 inches

If key artwork alternative, image provided must have one focal point (i.e., not too crowded/busy - 1 - 2 people max).

Lower 1/8 of portrait art is clear of important details

### **Naming**

Use EIDR if available

If no EIDR available, please refer to the naming conventions agreed upon by the Studio/Network and Digital Distributor

### **Foreign Releases**

Need the localized name

Would prefer art used in the local territories

Use "series" instead of "season" when applicable.

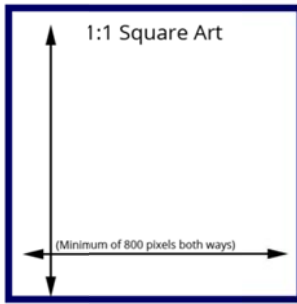
### **No key art available**

Choose an image to be consistent across platforms such as a still frame or images from other promotional materials

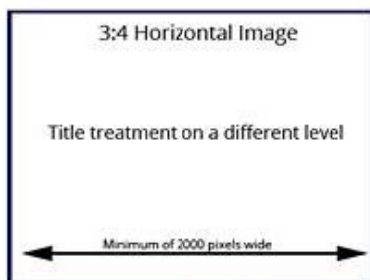
### **Additional Images**

Minimum of two stills in each of above sizes that are relevant to the audience (i.e., violent vs. conservative image.)





## HORIZONTAL (16:9 and 3:4) IMAGES



## **The EMA Digital Supply Chain Image File Work Group**

Meg Campbell, Google/YouTube – Work Group Chair

Best Buy – Randy Pierce

Google/YouTube – Matt Darby and Mark Fleming

Microsoft – Ellen Kim

Netflix – Patrick McGowan and Nick Nelson

Vudu – Adam Simpson and Heather Carrie

EMA Staff – Mark Fisher, Jen Lane Burnell, Steve Apple

For further information, please contact Mark Fisher, EMA, [mfisher@entmerch.org](mailto:mfisher@entmerch.org)  
or Jennifer Lane Burnell, EMA, [jlane@entmerch.org](mailto:jlane@entmerch.org).



entertainment merchants association

16530 Ventura Blvd., Suite 400

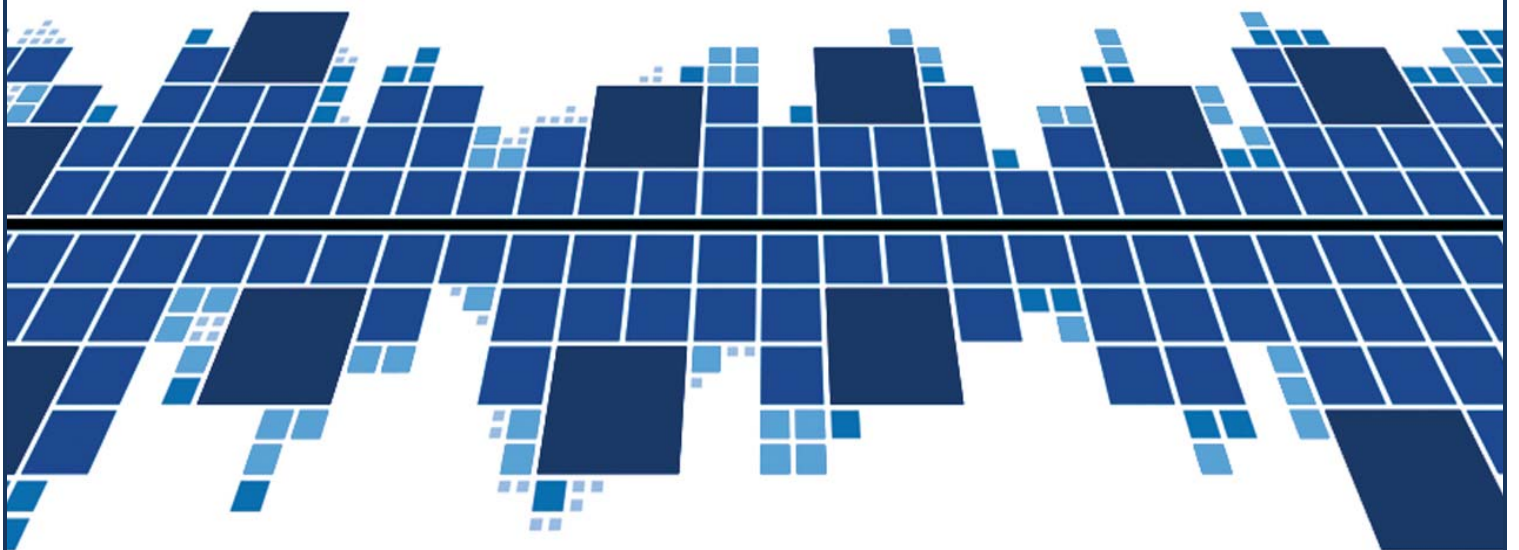
Encino, CA 91436

818.385.1500

[www.entmerch.org](http://www.entmerch.org)

# **Best Practices & Standards For the Delivery of Avails for Digital Audio-Visual Distribution**

January 8, 2013







## INTRODUCTION

EMA's Digital Supply Chain work groups and committees have been working diligently on defining a standard metadata structure, recommended mezzanine file specs, a set of recommended image files, and captioning issues.

One pain-point that has not yet been addressed may be a fairly simple one to resolve – the inconsistency and inaccuracy of avails information.

### Why is it needed?

Each studio currently provides avails to retailers in different formats (e.g., in the body of emails, Excel spreadsheets, PDFs, and sometimes even as image files) typically via email. These avails are arguably the most important data we handle:

- Importance to Studios:
  - Dates need to be accurately communicated so a title doesn't go live earlier than planned, and comes down in time so not to violate terms with highly valued Pay TV partners (e.g., HBO, Showtime) or other licensees/licensors they have agreements with.
  - An inaccurate start date that delays the release of a title will result in missed revenue opportunities.
- Importance to Retailers:
  - Avail notices kick off the scheduling and production workflow (ordering and fulfillment); a missed avail notice can delay the release of a title, resulting in missed revenue opportunities.
  - Going live too early or keeping a title up too long (outside of avail window) can negatively impact relationships with the studios.
  - Delay in release of film creates a negative consumer experience with platform, especially for high profile titles where home video release dates are advertised.
- No standards exist across any of the studios or retailers
  - Studios: notices can come in a variety of formats (e.g., body of email, Excel spreadsheets, PDFs) and often require manual work for a retailer to ingest, bringing with it a high risk of entry error and/or a high cost in resource needs (e.g., in auditing all entries).
  - Retailers: many have their own templates or portals, which are a burden on studios to complete for each. Studios are then resistant to using retailer templates because of resource limitations and liability; the often manual conversion to templates -- and inherent risk of entry error -- would be pushed to the studio's side.
- Notices often come at inconsistent intervals; few studios send notice on a consistent/predictable schedule.

Retail members of the EMA have developed a proposed standard set of avails in an organized set of accessible formats (both XML and Excel) to meet everyone's needs.





# Content Availability Metadata (Avails)

---

Ref: TR-META-AVAIL  
Version: 1.0  
Date: January 3, 2013

# EMA Content Availability Metadata (Avails)

## CONTENTS

1	Introduction .....	1
1.1	Document Organization .....	1
1.2	Document Notation and Conventions .....	1
1.2.1	XML Conventions .....	1
1.2.2	General Notes .....	2
1.3	Normative References .....	3
1.4	Informative References.....	3
1.5	XML Namespaces .....	3
1.6	Identifiers .....	4
1.7	Status .....	4
2	Avail Information .....	5
2.1	Avail List .....	5
2.2	Avail.....	5
2.2.1	AvailDisposition-type .....	6
2.2.2	AvailAsset-type.....	7
2.2.3	AvailTrans-type.....	10
2.2.4	Money-type and NVPairMoney-type (should be Common Metadata).....	14
3	Rules for Spreadsheet Encoding .....	16
3.1	Mapping Spreadsheet to XML Document.....	16
3.2	Mapping XML Document to Spreadsheet.....	17

**NOTE:** No effort is being made by EMA, the EMA Digital Council or Motion Picture Laboratories to in any way obligate any market participant to adhere to the Common Metadata or EMA Metadata. Whether to adopt the Common Metadata and/or EMA Metadata in whole or in part is left entirely to the individual discretion of individual market participants, using their own independent business judgment. Moreover, EMA, the EMA and Motion Picture Laboratories each disclaim any warranty or representation as to the suitability of the Common Metadata and/or EMA Metadata for any purpose, and any liability for any damages or other harm you may incur as a result of subscribing to this Metadata.



---

## REVISION HISTORY

Version	Date	Description
1.0	January 3, 2013	Original Version

## 1 INTRODUCTION

The Entertainment Merchant's Association (EMA) has defined the means to delivery Content Availability (Avails) data. 'Avails' is an industry term for business information regarding the availability of assets to be offered. It includes information such as region of available, times of available and business terms. This document was developed by the EMA Digital Council with the objective of standardizing the metadata communication from content providers to digital retailers.

This document defines the EMA Avails version 1.0.

The document describes encoding for Avails data in both spreadsheet form and in XML form. Although spreadsheets may serve an interim purpose, migration to XML is encouraged.

EMA Avails Metadata builds upon Media Entertainment Core (MEC) Metadata, and also Common Metadata developed by Motion Picture Laboratories, EMA, DEG and others.

### 1.1 Document Organization

This document is organized as follows:

1. Introduction—Provides background, scope and conventions
2. Avails—The definition of Avails data. This includes encoding information that applies to both spreadsheets and XML; and the XML definition.
3. Rules for Spreadsheet Encoding – Information on using Section 2 definitions within spreadsheets. Also, information on mapping between spreadsheets and XML.

### 1.2 Document Notation and Conventions

#### 1.2.1 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.

##### 1.2.1.1 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 initialLowercaseAttribute.
- XML structures are formatted as Courier New, such as `md:rightstoken`
- Names of both simple and complex types are followed with “-type”

## 1.2.1.2 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 or type
- 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 descendents) 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.

## 1.2.2 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.

## 1.3 Normative References

- [CM] TR-META-CM MovieLabs Common Metadata, version 2.0, <http://www.movielabs.com/md/md>
- [MECMD] DEG-EMA Media Entertainment Core Metadata, version 2.0, <http://www.movielabs.com/md/mec>
- [RFC2141] R. Moats, *RFC 2141, URN Syntax*, May 1997, <http://www.ietf.org/rfc/rfc2141.txt>
- [RFC3629] Yergeau, F., et al, *RFC 3629, UTF-8, a transformation format of ISO 10646*, November, 2003. <http://www.ietf.org/rfc/rfc3629.txt>
- [RFC3986] Berners-Lee, T., et al, RFC 3986, Uniform Resource Identifier (URI): Generic Syntax, January 2005, <http://www.ietf.org/rfc/rfc3986.txt>
- [RFC5646] Philips, A, et al, *RFC 5646, Tags for Identifying Languages*, IETF, September, 2009. <http://www.ietf.org/rfc/rfc5646.txt>
- [IANA-LANG] IANA Language Subtag Registry. <http://www.iana.org/assignments/language-subtag-registry>
- [ISO3166-1] Codes for the representation of names of countries and their subdivisions -- Part 1: Country codes, 2007.
- [ISO3166-2] ISO 3166-2:2007 Codes for the representation of names of countries and their subdivisions -- Part 2: Country subdivision code
- [ISO4217] Currency shall be encoded using ISO 4217 Alphabetic Code. [http://www.iso.org/iso/currency\\_codes\\_list-1](http://www.iso.org/iso/currency_codes_list-1)
- [ISO8601] ISO 8601:2000 Second Edition, *Representation of dates and times, second edition*, 2000-12-15.
- [CEA766] ANSI/CEA-766-C, U.S. and Canadian Rating Region Tables (RRT) and Content Advisory Descriptors for Transport of Content Advisory Information Using ATSC Program and System Information Protocol (PSIP). April 2008.

## 1.4 Informative References

- [RFC4647] Philips, A., et al, *RFC 4647, Matching of Language Tags*, September 2006. <http://www.ietf.org/rfc/rfc4647.txt>
- European Broadcast Union, Tech 3295 – P\_META Metadata Library, [http://www.ebu.ch/en/technical/metadata/specifications/notes\\_on\\_tech3295.php](http://www.ebu.ch/en/technical/metadata/specifications/notes_on_tech3295.php)

## 1.5 XML Namespaces

This document refers to the following XML namespaces:

- md: Common Metadata corresponding with Common Metadata.

- mdmec: Media Entertainment Core Metadata. Note that mdmec: references md: schemas
- avails: includes Avails data. Note that avails: references md: and mdmec: schemas

## 1.6 Identifiers

Identifiers must be universally unique. Recommended identifier schemes may be found in Common Metadata and in UltraViolet Content Metadata.

## 1.7 Status

This specification is completed and ready for pilot implementation. Although tested, we anticipate that additional implementation experience will yield recommendation for changes. Implementers should anticipate one or more revisions. Reasonable measures will be taken to ensure changes are backwards compatible.

## 2 AVAIL INFORMATION

The top level element for Avails are `Avail` and `AvailList`. The top-level XML type for Avails are `Avail-type` and `AvailList-type`.

### 2.1 Avail List

An Avail List contains on or more Avials.

Element	Attribute	Definition	Value	Card.
AvailList		Element for an Avail List	avails:AvailList-type	

Element	Attribute	Definition	Value	Card.
AvailList-type				
Avail		An Avail	avails:Avail-type	1..n

### 2.2 Avail

The Avail element is defined as follows:

Element	Attribute	Definition	Value	Card.
Avail		Element continuing a single Avail	avail:Avail-type	

The Avail-type complex type is defined as follows:

Element	Attribute	Definition	Value	Card.
Avail-type				
Disposition		Information about the Avails message such as whether it is a new Avail or if it replaces a previous Avail message.	avails:AvailDisposition-type	

Licensor		The entity issuing the Avail	mdmec:Publisher-type	
Asset		Each instance defines an asset subject to the Avail instructions	avails:AvailAsset-type	1..n
Transaction		Each instance includes transaction information regarding the Avail	avails:AvailTransaction-type	1..n
OfferingContentStructure		Description of relationship of the asset to each other in the form of a Common Metadata Compliation.	md:CompObj-type	0..1

## 2.2.1 AvailDisposition-type

Element	Attribute	Definition	Value	Card.
AvailDisposition-type				
EntryType		Indication of whether this Avail is new, update or deletion.	xs:string	
EntryID		An identifier unique to the Licensor that identifies this Avail. EntryID is used to match Avails for Update and Delete operations. It can also be used by respective parties to refer to the Avail.	md:id-type	0..1
IssueDate		Date this Avail was issued. If necessary, recipients can use IssueDate to reconstruct the order of issuance. Although this may be xs:gYear only or xs:date, it is strongly recommended that the xs:dateTime form be used.	md:YearDateOrTime-type	0..1
ReplacesEntryID		An EntryID in an Avail previously sent. Used to match for purposes of updating or deleting.	md:id-type	(optional choice)
ReplacesEntryDate		An EntryDate in an Avail previously sent. Used to match for purposes of updating or deleting.	md:YearDateOrTime-type	(optional choice)
<any>		Any other element	any ##other	0..n

EntryType shall have one of the following values:

- “Create” – Creates a new Avail.
- “Update” – Updates a matching Avail. See note below on matching. This Avail will replace the previous Avail in its entirety.
- “Delete” – Deletes the matching Avail. See note below on matching.
- “Other” – The recipient should evaluate the current Avail against existing Avails and determine whether this Avail is new or an update. It is recommended that “Update” be used instead of “Other”.

An Avail matches an earlier Avail if the new Avail’s ReplacesEntryID matches the earlier Avail’s EntryID, or the new Avail’s ReplacesEntryDate matches the earlier Avail’s EntryDate.

## 2.2.2 AvailAsset-type

Element	Attribute	Definition	Value	Card.
AvailAsset-type				
	contentID	Asset Identifier. This should be an EIDR.	md:ContentID-type	
WorkType		Work type as enumerated in Common Metadata, and repeated below.	xs:string	
TitleInternalAlias		Title used by involved parties to refer to this content.	xs:string	
ProductID		An identifier mutually agreed upon by sender and recipient. ProductID must be unique within a licensor. It is preferable that it be globally unique, such an EIDR.	xs:string	0..n
Metadata		Metadata describing Asset	avails:AvailMetadata-type	
SeriesMetadata		Additional metadata describing series information, such as seasons and series. This shall only be included if the asset is part of a series (e.g., an episode)	avails:AvailSeriesMetadata-type	0..1



WorkType shall be enumerated to one of the following (categories are to support the definition, but are not included in the enumeration).

Music related:

- 'Album' – A collection of songs
- 'Song'
- 'Music Video' – Music Video, not 'Performance'
- 'Ring Tone'
- 'Other Music'

Film related:

- 'Movie' – A full length movie regardless of distribution (e.g., theatrical, TV, direct to disc, etc.) and content (e.g., includes documentaries).
- 'Short' – a film of length shorter than would be considered a feature film.

TV, web and mobile related:

- 'Series' – a show that might span one or more seasons or might be a miniseries.
- 'Season' – a season of a Series. It will contain one more episodes.
- 'Episode' – an episodes of a season or miniseries. A pilot is also an episode. If episode is a 'webisode', 'mobisode' or other specialized sequence, it should be noted in Keywords.
- 'Non-episodic Show' – TV or other show that is non-episodic; for example, sports and news.
- 'Promotion' – promotional material associated with media. This includes teasers, trailers, electronic press kits and other materials. Promotion is a special case of 'Ad'.
- 'Ad' – any form of advertisement including TV commercials, informercials, public service announcements and promotions not covered by 'Promotion'. This does not include movie trailers and teasers even though they might be aired as a TV commercial.

Other:

- 'Excerpt' – An asset that consists primarily of portion or portions of another work or works; for example, something having the 'isclipof' or 'iscompositeof' relationship.
- 'Supplemental' – Material designed to supplement another work. For example, and extra associated with a Movie for a DVD.

- ‘Collection’ – A collection of assets not falling into another category. For example, a collection of movies.
- ‘Franchise’ – A collection or combination of other types, for example, a franchise might include multiple TV shows, or TV shows and movies.

## 2.2.2.1 AvailMetadata-type

Element	Attribute	Definition	Value	Card.
AvailMetadata-type				
TitleDisplayUnlimited		Display title, no length limit. Same as TitleDisplayUnlimited in Common Metadata.	xs:string	
RunLength		Total run time. Same as RunLength in Common Metadata.	xs:duration	0..1
AltIdentifier		Other identifiers referring to the same asset. Same as AltIdentifier in CommonMetadata.	md:ContentIdentifier-type	0..n
ReleaseHistory		History of release such as air dates or DVD release information. Defined in Common Metadata, 4.1.1.	md:ReleaseHistory-type	0..n
<any>		Any other element	any ##other	0..n

## 2.2.2.2 AvailSeriesMetadata-type

Element	Attribute	Definition	Value	Card.
AvailSeriesMetadata-type				
SeriesID		Identifier for Series. Preferably an EIDR.	md:id-type	
SeasonID		Identifier for Season. Preferably an	md:id-type	

		EIDR.		
SeriesTitle		Title for series in language mutually agreed upon by sender and receiver. Same as Core Metadata TitleInternalAlias	xs:string	
SeasonTitle		Title for season. Same as Common Metadata TitleDisplayUnlimited for WorkType 'Season.	xs:string	
LocalSeriesTitle		Local series title, if applicable. Same as Common Metadata TitleDisplayUnlimited for WorkType 'Series'	xs:string	0..n
	language	Language for local series title	xs:language	
LocalSeasonName		Local season title, if applicable. Same as Common Metadata TitleDisplayUnlimited for WorkType 'Season.	xs:string	0..n
	language	Language for local series title	xs:language	
SeasonNumber		Season number as defined in Common Metadata. Parties should agree upon which numbering scheme to use.	md:ContentSequenceInfo-type	
SeasonEpisodeCount		Number of episodes in this season.	xs:positiveInteger	0..1
SeriesAltIdentifier		Other identifiers for the series.	md:ContentIdentifier-type	0..n
SeasonAltIdentifier		Other identifiers for the season.	md:ContentIdentifier-type	0..n
<any>		Any other element	any ##other	0..n

## 2.2.3 AvailTrans-type

Element	Attribute	Definition	Value	Card.
AvailTrans-type				

# Content Availability Metadata (Avails)

Type		Type of transaction. See below.	xs:string	
Description		A free-form description of the transaction.	xs:string	
Locale		Region or regions where transaction applies. Default is worldwide. Note that if both Locale and LocaleExcluded are absent, default is worldwide.	md:Region-type	0..n
Language		Language or languages to which transaction applies. If absent, then all languages is assumed.	xs:language	0..n
LocaleExcluded		Region or regions where transaction does not apply. Default is nowhere, and Locale takes precedence.	md:Region-type	0..n
LicenseRightsDescription		Description of License or Rights granted. See below.	xs:string	
FormatProfile		Indicates the format profile covered by the transaction. This typically refers to HD, SD or 3D.	xs:string	
Terms		Terms described in pre-defined values.	avails:AvailTerms-type	0..1
OtherTerms		Terms described as name/value pairs.	md:NVPair-type	0..1
OtherTerms		Monetary terms described as name/value pairs.	md:NVPairMoney-type	0..1
OtherInstructions		Any other instructions. Free text.	xs:string	0..1
Start		Start of terms. If Start and ContStart are absent, terms begin immediately.	xs:dateTime	(optional choice with ContStart)
CondStart		Conditional Start of terms	avails:AvailTransCondDate-type	(optional choice with Start)

End		End of terms. If End and CondEnd are absent, terms continue indefinitely.	xs:dateTime	(optional choice with CondEnd)
CondEnd		Conditional ending period	avails:AvailTransCondDate-type	(optional choice with End)

Type should have one of the following values, although additional values may be used by agreement between sender and receiver:

- ‘EST’ (Electronic Sell Through)
- ‘VOD’ (Video on Demand) – Download or streaming based on individual transactions (e.g., payment per use).
- ‘SVOD’ (Subscription VOD) – Streaming on a subscription service

Note that any of these models can be paid or free.

LicenseRightsDescription should have one of the following values:

- ‘New Release’ – New release
- ‘Library’ – Catalog title
- ‘Mega-Library’ – High value library
- ‘DD-Theatrical’ – EST, VOD or Subscription availability, Day and Date withTheatrical
- ‘Pre-Theatrical’ – EST, VOD or Subscription availability prior to theatrical availability
- ‘DD-DVD’ – Day and Date DVD
- ‘Early EST’ – EST prior to DVD availability
- ‘Preorder EST’ – preorder EST prior to DVD availability (order, but not download or play)
- ‘Early VOD’ – VOD prior to DVD availability, also Preorder VOD
- ‘Preorder VOD’ – preorder VOD prior to DVD availability (order, but not download or play)
- ‘DTV’ – Direct to Video

FomatProfile should have one of the following values

- ‘HD’ – High Definition

- ‘SD’ – Standard Definition
- ‘3D’ – 3D, non-specific of resolution
- ‘3DHD’ – 3D High Definition
- ‘3DSD’ – 3D Standard Definition
- ‘HFR’ – HD High Frame Rate
- ‘3DHFR’ – 3D High Frame Rate
- ‘4K’ – 4K (4096x2160) format or 4xHD (3840x2160)
- ‘3D4K’ – 3D 4K

### 2.2.3.1 AvailTransCondDate-type

Element	Attribute	Definition	Value	Card.
TransInfo-type				
Event		The event to which this condition is tied	xs:string	0..1
Condition		Indication of before, after, etc.	xs:string	
Locale		Locale of the condition	md:Region-type	0..1
Lag		Indication of how much before or after the event. This shall always be positive and the direction is assumed from the Condition.	xs:duration	0..1

Event may have any value as listed under Release Information Encoding as described in the Common Metadata Specification.

The following are accepted values for Condition

- ‘before’ – indicates Lag before Event
- ‘after’ – indicates Lag after Event
- ‘simultaneous’ – indicates it happens at the same time. Lag should not be included, but ignored if it is.

### 2.2.3.2 AvailTerms-type

Element	Attribute	Definition	Value	Card.
AvailTerms-type				
Tier		Offering Tier. This is typically a reference to a tier in a contract.	xs:string	0..1
RentalDuration		Duration of rental period.	xs:duration	0..1
WatchDuration		How long user has to complete viewing once started.	xs:duration	0..1
WSP		Wholesale price	md:Money-type	0..1
MSRP		Manufactures Suggested Retail Price	md:Money-type	0..1
SeasonWSP		Wholesale Price for an entire season.	md:Money-type	0..1
CaptionIncluded		Are captions included in encoding. 'true' means yes.	xs:boolean	0..1
CaptionRequired		Are captions required anywhere in the regions specified for the transaction. 'true' means yes.	xs:boolean	0..1
<any>		Any other element	any ##other	0..n

## 2.2.4 Money-type and NVPairMoney-type (should be Common Metadata)

Element	Attribute	Definition	Value	Card.
Money-type				
	currency	Currency as expressed in ISO 4217 Currency Alphabetic Code. For example, 'USD' for US Dollars.	xs:string	
Value		Value	xs:decimal	

[ISO4217] typically allows two or three digits after the decimal. However, Value in this element may have as many decimal places as necessary.

NVPairMoney-type is like NVPair-type except the Value is currency-based.

# Content Availability Metadata (Avails)

---

Element	Attribute	Definition	Value	Card.
NVPairMoney-type				
Name		Identification of the parameter being specified	xs:string	
Value		Value specified for Name.	md:Money-type	



## 3 RULES FOR SPREADSHEET ENCODING

The Excel structure is a subset of what can be encoded in the XML structure. This section defines rules for translating between schema and spreadsheet.

Note that spreadsheets may be transmitted as Microsoft Excel spreadsheets or common separated values (CSV) files.

### 3.1 Mapping Spreadsheet to XML Document

Spreadsheet		XML	Mapping
Licensor	Display Name	//Licensor/DisplayName	1:1
AvailTrans	Language	//Transaction/Language	1:1
AvailTrans	Locale	//Transaction/Locale	1:1
Avail Asset	WorkType	//Asset/WorkType	1:1
Disposition	EntryType	//Disposition/EntryType	1:1
Avail Asset	TitleInternalAlias	//Asset/TitleInternalAlias	1:1
AvailMetadata	TitleDisplayUnlimited	//Asset/Metadata/TitleDisplayUnlimited	1:1
AvailTrans	LicenseRightsDescription	//Transaction/LicenseRightsDescription	1:1
AvailTrans	(License) Type	//Transaction/Type	1:1
AvailTrans	FormatProfile	//Transaction/FormatProfile	1:1
AvailTrans	Start	//Transaction/Start	1:1
AvailTrans	End	//Transaction/End	1:1
AvailTrans	Description	//Transaction/Description	1:1
AvailTrans	Other Terms	//Transaction/OtherTerms	Name/Value pairs in parentheses, separated by semicolon ( <i>name;value</i> )
AvailTrans	Other Instructions	//Transaction/OtherInstructions	1:1
Avail Asset	Content ID	//Asset/@contentID	1:1
Avail Asset	Product ID	//Asset/ProductID	1:1
AvailMetadata	AltID	//Asset/Metadata/AltIdentifier	First instance
AvailMetadata	Release History (Original)	//Asset/ReleaseHistory/Date	Where ReleaseType='original' and DistrTerritory matches
AvailMetadata	Release History (DVD)	//Asset/ReleaseHistory/Date	Where ReleaseType='DVD' and DistrTerritory matches
AvailTerms	Rental Duration	//Transaction/Terms/RentalDuration	1:1
AvailTerms	Watch Duration	//Transaction/Terms/WatchDuration	1:1
AvailTerms	WSP	//Transaction/Terms/WSP	1:1

# Content Availability Metadata (Avails)

AvailTerms	Tier	//Transaction/Terms/Tier	
AvailTerms	MSRP	//Transaction/Terms/MSRP	1:1
AvailTerms	CaptionIncluded	//Asset/Metadata/CaptionIncluded	1:1
AvailTerms	Caption Required	//Asset/Metadata/CaptionRequired	1:1
AvailTerms	Any		
AvailMetadata	Total Run Time	//Asset/Metadata/RunLength	1:1

## 3.2 Mapping XML Document to Spreadsheet

XML	Spreadsheet		Mapping
//Disposition/EntryType			None
//Disposition/EntryType	Disposition	EntryType	1:1
//Disposition/IssueDate			None
//Disposition//ReplacesEntryid			None
//Disposition/ReplacesEntryDate			None
//Licensor/@organizationID			None
//Licensor/@idType			None
//Licensor/@retailerSpecificID			None
//Licensor/DisplayName	Licensor	Display Name	1:1
//Licensor/SortName			None
//Licensor/AlternateName			None
//Licensor/ContactInfo			None
//Asset/@contentID	Avail Asset	Content ID	1:1
//Asset/WorkType	Avail Asset	WorkType	1:1
//Asset/TitleInternalAlias	Avail Asset	TitleInternalAlias	1:1
//Asset/ProductID	Avail Asset	Product ID	1:1
//Asset/Metadata/AltIdentifier	AvailMetadata	AltID	First instance
//Asset/Metadata/TitleDisplayUnlimited	AvailMetadata	TitleDisplayUnlimited	1:01
//Asset/Metadata/RunLength	AvailMetadata	Total Run Time	1:01

# Content Availability Metadata (Avails)

//Asset/ReleaseHistory/Date	AvailMetadata	Release History (Original), Release History (DVD)	ReleaseHistory/Date maps where ReleaseType='original' and ReleaseType='DVD'
//AssetReleaseHistory/ReleaseType			Maps in cases above
//AssetReleaseHistory/DistrTerritory			Maps in cases above
//AssetReleaseHistory/Description			None
//Asset/ReleaseOrg			None
//Asset/Metadata/CaptionIncluded	AvailTerms	CaptionIncluded	1:1
//Asset/Metadata/CaptionRequired	AvailTerms	Caption Required	1:1
//Asset/SeriesMetadata			Future
//Transaction/Description	AvailTrans	Description	1:1
//Transaction/Locale	AvailTrans	Locale	Only first instance. Must have a new row for each combination.
//Transaction/Language	AvailTrans	Language	
//Transaction/LocaleExcluded			This cannot be represented. If necessary, all other Locales included in Locale.
//Transaction/LicenseRightsDescription	AvailTrans	LicenseRightsDescription	1:1
//Transaction/FormatProfile	AvailTrans	FormatProfile	1:1
//Transaction/Type	AvailTrans	(License) Type	1:1
//Transaction/Terms/Tier	AvailTrans	Tier	1:1
//Transaction/Terms/RentalDuration	AvailTerms	Rental Duration	1:1
//Transaction/Terms/WatchDuration	AvailTerms	Watch Duration	1:1
//Transaction/Terms/WSP	AvailTerms	WSP	1:1
//Transaction/Terms/MSRP	AvailTerms	MSRP	1:1
//Transaction/Terms/SeasonWSP			Future
//Transaction/OtherTerms	AvailTrans	Other Terms	Name/Value pairs in parentheses, separated by semicolon ( <i>name;value</i> )
//Transaction/OtherFinanceTerms			None

# Content Availability Metadata (Avails)

//Transaction/OtherInstructions	AvailTrans	Other Instructions	1:1
//Transaction/Start	AvailTrans	Start	1:1
//Transaction/CondStart			None
//Transaction/End	AvailTrans	Start	1:1
//Transaction/CondEnd			None
//OfferingContentStructure			None
//CoreMetadata			None

This XML file does not appear to have any style information associated with it. The document tree is shown below.

---

```
<!--
  edited with XMLSpy v2013 sp1 (x64) (http://www.altova.com) by Craig Seidel
  (MovieLabs)
-->
<xs:schema xmlns:avails="http://www.movielabs.com/schema/avails/v1.0/avails"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:md="http://www.movielabs.com/schema/md/v2.0/md"
  xmlns:mdmec="http://www.movielabs.com/schema/mdmec/v2.0"
  targetNamespace="http://www.movielabs.com/schema/avails/v1.0/avails"
  elementFormDefault="qualified" attributeFormDefault="unqualified" version="1.0">
  <xs:import namespace="http://www.movielabs.com/schema/md/v2.0/md"
    schemaLocation="http://www.movielabs.com/schema/md/v2.0/md-v2.0.xsd"/>
  <xs:import namespace="http://www.movielabs.com/schema/mdmec/v2.0"
    schemaLocation="http://www.movielabs.com/schema/mdmec/v2.0/mdmec-v2.0.xsd"/>
  <!-- EMA CONTENT AVAILABILITY METADATA (Avails) -->
  <!-- version 1.0 -->
  <xs:complexType name="TransCondDate-type">
    <xs:sequence>
      <xs:element name="Event" type="xs:string"/>
      <xs:element name="Condition" type="xs:string"/>
      <xs:element name="Locale" type="md:Region-type"/>
      <xs:element name="Lag" type="xs:duration"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="AvailsPublisher-type">
    <xs:complexContent>
      <xs:extension base="md:OrgName-type">
        <xs:sequence>
          <xs:element name="ContactInfo" type="md:ContactInfo-type"/>
        </xs:sequence>
        <xs:attribute name="retailerSpecificID" type="xs:string"/>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
  <xs:complexType name="AvailDisposition-type">
    <xs:sequence>
      <xs:element name="EntryType" type="xs:string"/>
      <xs:element name="EntryID" type="md:id-type" minOccurs="0"/>
      <xs:element name="IssueDate" type="md:YearDateOrTime-type" minOccurs="0"/>
      <xs:choice>
        <xs:element name="ReplacesEntryID" type="md:id-type" minOccurs="0"/>
        <xs:element name="ReplacesEntryDate" type="md:YearDateOrTime-type"
          minOccurs="0"/>
      </xs:choice>
      <xs:any namespace="##other" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="AvailSeriesMetadata-type">
    <xs:sequence>
      <xs:element name="SeriesID" type="md:id-type"/>
      <xs:element name="SeasonID" type="md:id-type"/>
      <xs:element name="SeriesTitle" type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
</xs:schema>
```

```

<xs:element name="SeasonTitle" type="xs:string"/>
<xs:element name="LocalSeriesTitle" minOccurs="0" maxOccurs="unbounded">
  <xs:complexType>
    <xs:simpleContent>
      <xs:extension base="xs:string">
        <xs:attribute name="language" type="xs:language"/>
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>
</xs:element>
<xs:element name="LocalSeasonName" minOccurs="0">
  <xs:complexType>
    <xs:simpleContent>
      <xs:extension base="xs:string">
        <xs:attribute name="language" type="xs:language"/>
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>
</xs:element>
<xs:element name="SeasonNumber" type="md:ContentSequenceInfo-type"/>
<xs:element name="SeasonEpisodeCount" type="xs:positiveInteger" minOccurs="0"/>
<xs:element name="SeriesAltIdentifier" type="md:ContentIdentifier-type"
minOccurs="0" maxOccurs="unbounded"/>
<xs:element name="SeasonAltIdentifier" type="md:ContentIdentifier-type"
minOccurs="0" maxOccurs="unbounded"/>
<xs:any namespace="##other" minOccurs="0" maxOccurs="unbounded"/>
</xs:sequence>
</xs:complexType>
<xs:complexType name="AvailMetadata-type">
  <xs:sequence>
    <xs:element name="AltIdentifier" type="md:ContentIdentifier-type" minOccurs="0"
maxOccurs="unbounded"/>
    <xs:element name="TitleDisplayUnlimited">
      <xs:complexType>
        <xs:simpleContent>
          <xs:extension base="xs:string"/>
        </xs:simpleContent>
      </xs:complexType>
    </xs:element>
    <xs:element name="ReleaseHistory" type="md:ReleaseHistory-type" minOccurs="0"
maxOccurs="unbounded"/>
    <xs:element name="CaptionIncluded" type="xs:boolean" minOccurs="0"/>
    <xs:element name="CaptionRequired" type="xs:boolean" minOccurs="0"/>
    <xs:any namespace="##other" minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="AvailTerms-type">
  <xs:sequence>
    <xs:element name="LicenseType" type="xs:string" minOccurs="0"/>
    <xs:element name="RentalDuration" type="xs:duration" minOccurs="0"/>
    <xs:element name="WatchDuration" type="xs:duration" minOccurs="0"/>
    <xs:element name="WSP" type="md:Money-type" minOccurs="0"/>
    <xs:element name="MSRP" type="md:Money-type" minOccurs="0"/>
    <xs:element name="SeasonWSP" type="md:Money-type" minOccurs="0"/>
    <xs:any namespace="##other" minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>

```

```

<xs:complexType name="AvailTrans-type">
  <xs:sequence>
    <xs:element name="LicenseType" type="xs:string"/>
    <xs:element name="Description" type="xs:string"/>
    <xs:element name="Locale" type="md:Region-type" minOccurs="0"
maxOccurs="unbounded">
      <xs:annotation>
        <xs:documentation>work Intended Audience Region</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="LocaleExcluded" type="md:Region-type" minOccurs="0"
maxOccurs="unbounded">
      <xs:annotation>
        <xs:documentation>work Intended Audience Region</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="LicenseRightsDescription" type="xs:string"/>
    <xs:element name="FormatProfile" type="xs:string"/>
    <xs:element name="Terms" type="avails:AvailTerms-type" minOccurs="0"/>
    <xs:element name="OtherTerms" type="md:NVPair-type" minOccurs="0"
maxOccurs="unbounded"/>
    <xs:element name="OtherFinanceTerms" type="md:NVPairMoney-type" minOccurs="0"
maxOccurs="unbounded"/>
    <xs:element name="OtherInstructions" type="xs:string" minOccurs="0"/>
    <xs:choice>
      <xs:element name="Start" type="xs:dateTime" minOccurs="0"/>
      <xs:element name="CondStart" type="avails:TransCondDate-type" minOccurs="0"/>
    </xs:choice>
    <xs:choice>
      <xs:element name="End" type="xs:dateTime" minOccurs="0"/>
      <xs:element name="CondEnd" type="avails:TransCondDate-type" minOccurs="0"/>
    </xs:choice>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="AvailAsset-type">
  <xs:sequence>
    <xs:element name="Type" type="xs:string"/>
    <xs:element name="TitleInternalAlias" type="xs:string"/>
    <xs:element name="ProductID" type="xs:string" minOccurs="0"
maxOccurs="unbounded"/>
    <xs:element name="Metadata" type="avails:AvailMetadata-type"/>
    <xs:element name="SeriesMetadata" type="avails:AvailSeriesMetadata-type"
minOccurs="0"/>
  </xs:sequence>
  <xs:attribute name="contentID" type="md:ContentID-type" use="required"/>
</xs:complexType>
<xs:complexType name="Avail-type">
  <xs:sequence>
    <xs:element name="Disposition" type="avails:AvailDisposition-type"/>
    <xs:element name="Licensor" type="mdmec:Publisher-type"/>
    <xs:element name="Asset" type="avails:AvailAsset-type" maxOccurs="unbounded"/>
    <xs:element name="Transaction" type="avails:AvailTrans-type"
maxOccurs="unbounded"/>
    <xs:element name="OfferingContentStructure" type="md:CompObj-type"
minOccurs="0"/>
    <xs:element name="CoreMetadata" type="mdmec:CoreMetadata-type" minOccurs="0"
maxOccurs="unbounded"/>
  </xs:sequence>

```

```
</xs:sequence>
</xs:complexType>
<xs:complexType name="AvailList-type">
  <xs:sequence>
    <xs:element name="Avail" type="avails:Avail-type" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>
<xs:element name="Avail" type="avails:Avail-type"/>
<xs:element name="AvailList" type="avails:AvailList-type"/>
</xs:schema>
```



**EMA Avails Excel Spreadsheet fields V1.0**

Licensor	Display Name
AvailTrans	Language
AvailTrans	Locale
Avail Asset	WorkType
Disposition	EntryType
Avail Asset	TitleInternalAlias
AvailMetadata	TitleDisplayUnlimited
AvailTrans	LicenseRightsDescription
AvailTrans	(License) Type
AvailTrans	FormatProfile
AvailTrans	Start
AvailTrans	End
AvailTrans	Description
AvailTrans	Other Terms
AvailTrans	Other Instructions
Avail Asset	Content ID
Avail Asset	Product ID
Avail Asset	Metadata
AvailMetadata	AltID
AvailMetadata	Release History (Original)
AvailMetadata	Release History (DVD)
AvailTerms	Rental Duration
AvailTerms	Watch Duration
AvailTerms	WSP
AvailTerms	Tier
AvailTerms	MSRP
AvailTerms	CaptionIncluded
AvailTerms	Caption Required
AvailTerms	Any
AvailMetadata	Total Run Time



entertainment merchants association

16530 Ventura Blvd., Suite 400

Encino, CA 91436

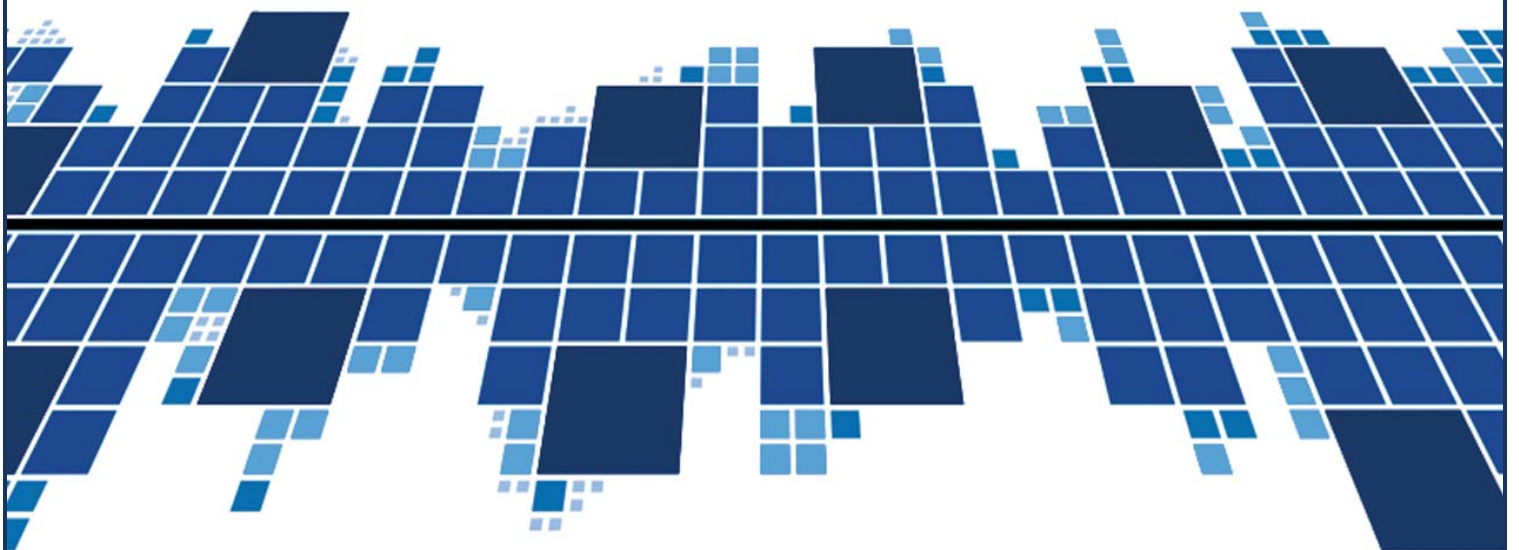
818.385.1500

[www.entmerch.org](http://www.entmerch.org)

# Best Practices for Closed Captioning of Internet Protocol-Delivered Video Programming

DRAFT

January 8, 2013







## **EMA’s Best Practices for Closed Captioning of Internet Protocol-Delivered Video Programming\***

The EMA Closed Captions Working Group was created to develop a better understanding of, and appropriate best practices for compliance with, the legal requirements imposed by federal law and regulation for closed captioning of Internet Protocol-delivered video programming and to identify other best practices for the conversion of television closed caption files for transmission over the Internet.

### **Certification If Captions Are Not Provided**

The Twenty-First Century Communications and Video Accessibility Act of 2010 and accompanying regulations promulgated by the Federal Communications Commission require that all full-length video delivered via the internet be able to be viewed with closed captions if the video airs on broadcast or cable television with closed captions on or after the applicable effective date. There are various effective dates for the requirement, which are dependent on whether the video programming is pre-recorded and whether it has been edited for the Internet:

- September 30, 2012, for all prerecorded programming that is not edited for Internet distribution;
- March 30, 2013, for all live and “near-live” programming [programming that is performed and recorded within 24 hours prior to its initial airing on television];

---

\*\*This document is a working draft and is subject to revision. Participation in the working group should not be considered endorsement of all of the draft recommendations.

- September 30, 2013, for all prerecorded programming that is edited for Internet distribution.

There are special rules for video programming that is already in the library of the video programming provider/distributor before it is shown on television with closed captions. Starting in March 2014, those videos must be captioned within 45 days of being shown on television with captions. The window is reduced to 30 days in 2015 and 15 days in 2016.

The regulations do not cover user-generated content (unless that content is included in the video programming as broadcast on television).

The content licensor (“video programming owner”) is to provide the closed captioning file to the online video service (“video programming distributor” or “video programming provider”). The content licensor and the online video service are to establish a mechanism for ongoing communication whether a particular video is covered by the closed captioning requirement, and the online video service must make a “good faith effort” to identify covered programming using that mechanism. An online video service is entitled to rely on a certification from the content provider that a particular video is not subject to the closed captioning requirement.

*Recommended Best Practice:*

*If a closed caption file is not provided for intended IP-delivered video programming, the video programming owner should include the following in both the avails and the metadata for that programming:*

*[Partner name] certifies that captions are not required for this video upload because:*

- This content has never aired on television in the U.S.*
- This content has only aired on television in the U.S. without captions.*
- This content has not aired on U.S. television with captions since September 30, 2012*

- This content does not consist of full-length video programming.*
- This content does not fall within a category of online programming that currently requires captions under FCC regulations (49 C.F.R. § 79.4(b)).*
- The FCC and/or U.S. Congress has granted an exemption from captioning requirements for this content.*

## **Preferred Captioning Formats**

In order for a broadcast video to be delivered over the Internet with closed captions, the closed caption file must be converted from the CEA-608 protocol used for television closed captions to a format suitable for Internet delivery (of which there are several), after which the captions can be edited to display properly. This conversion and editing can be done manually (extremely difficult and time-consuming), from scratch (very expensive), or by using software to extract and reformat the captioning data (preferred).

Even with closed captioning software, conversion can be challenging, especially when it involves a broadcast closed caption file in a legacy format. It is, therefore, desirable to have the closed caption file delivered in a format that is relatively easy to extract and reformat.

*Recommended Best Practice:*

*Delivery of closed caption files in SCC format (with a .scc file extension) is preferred at this time.*

- SCC contains exactly the CEA-608 information that is needed, is very concise, and is (almost entirely) unambiguous. It is also widely used and well understood by all captioning software.

Also accepted are the following TTML formats:

SMPTTE-TT (Recommended Practice 2052) (with a .xml file extension)

DFXP Full/Timed Text Markup Language (with a .dfxp file extension)

Please note, however:

- Using TTML markup carries some risk of inoperability, even with Recommended Practice 2052. We suggest only "pop-on" captions be used (regions pop1-4) to help simplify matters.
- Using the mechanism described in 5.10 of <https://www.smpete.org/sites/default/files/rp2052-10-2012.pdf> to "tunnel" CEA-608 data will result in unambiguous behavior. If we see this data, we may use it in preference to the other markup.
- Using arbitrary TTML as defined by <http://www.w3.org/TR/ttaf1-dfxp/> without reference to Recommended Practice 2052 is strongly discouraged.

Other formats are not preferred because either they are proprietary and not open source or they present technical issues (e.g., they do not support positioning).

## **Frame Rates**

Closed caption data files are separate from the video data files. Ideally, the caption frame rate should match the native frame rate of the source. However, they often do not, and synchronization of the two can be a problem.

Television in North America is generally broadcast at a standard rate of 29.97 frames per second (FPS). Internet video delivery, however, can support a variety of frame rate formats, and

a number of distributors of IP-delivered video programming require films and TV shows to be at a frame rate of 23.976 or 25 FPS.

These varying frame rate requirements mean that the closed caption files that were created for North American broadcast will not match the Internet video frame rate. As a result, the frame rate of the caption file must be reconfigured to the frame rate utilized by the Internet video content distributor (such as 23.976 FPS), and if necessary, the time code must be stretched or shrunk.

This can be a challenge for a number of reasons. In many cases, the caption file has SMPTE-based timestamps and fails to specify the frame rate. In such cases, one has to guess the frame rate until the correct frame rate is identified. In other cases, the video has been transcoded to a slightly different frame rate, or the captions were generated using a differently transcoded or edited version of the video.

To address this issue, some Internet video content distributors require the content provider to provide a closed caption data file that is already synchronized to the video data file. Others have developed processes to fix the caption files in-house.

*Recommended Best Practice:*

*If the Internet video content distributor does not require the closed caption data file to be already synchronized to the video data file, the closed caption data file may be submitted in any frame rate in which it was created, so long as the frame rate is clearly indicated in the file name, metadata, or code.*

**The EMA Closed Captions Working Group** (names and employers of Working Group Participants are for information only)



Amazon – Andy Bruncke, Steve Geiger, Joe Pavey

Best Buy – Brigit Lee

Google – Ken Harrenstien, Steve Maher

Microsoft Corp. – Michelle Burbridge, Laura Smith

MovieLabs – Craig Seidel

Netflix – Dae Kim

Rovi Corporation – David Hazlett, James Moore

Vudu – Aaron Martin, Adam Simpson

EMA Staff – Sean Bersell, Mark Fisher, Jen Lane

For further information, please contact Sean Bersell, EMA, [sbersell@entmerch.org](mailto:sbersell@entmerch.org), or Jennifer Lane Burnell, EMA, [jlane@entmerch.org](mailto:jlane@entmerch.org).