

JSON

Encoding Best Practice

This practice provides means to embed MDDF XML in JSON or to translate MDDF XML to JSON



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

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

REVISION HISTORY

Version	Date	Description
Draft	November 19, 2025	Initial draft for review
Draft 2	February 6, 2026	Clarifications of examples. Revised description of of top-level element.
1.0	TBD	Initial publication

1 OVERVIEW

While the official MDDF specifications are XML, there are situations where data needs to be delivered in JSON. In some cases, it is sufficient to embed the XML in JSON. In others, a JSON representation of the XML is needed. Both are addressed here

At some point in the future, we may create JSON schema.

2 EMBEDDING XML IN JAVASCRIPT

2.1 JSON Objects

The recommended approach is to use create a JSON object with a Value of type String.

For example,

```
{
  "mdmec": "<?xml version= \"1.0 \" encoding= \"UTF-8 \"?><xmlns:xsi=
  \"http://www.w3.org/2001/XMLSchema-instance \" xmlns:md=
  \"http://www.movie-labs.com/schema/md/v2.11/md \" xmlns:mdmec=
  \"http://www.movie-labs.com/schema/mdmec/v2.11 \"><mdmec:Basic ContentID=
  \"10.5240/0F1C-F1EC-F478-A1CC-1522-5 \">
  ... \"
}
```

2.2 Naming

Users may use whatever JSON names desired. The XML object's element defines its structure.

However, we recommend using the element name as the JSON name. For example, for MEC use 'mdmec' as the name.

2.3 Escaping XML characters not compatible with JSON strings

JSON has limitations on what can be represented in strings, so the following must be escaped

- Double quote (") – JSON terminates strings with quotation marks so any quotation marks in the XML must be escaped (\")
- Newline – Newline is considered white space and must be escaped to be preserved (\n). While generally not found in XML other whitespace characters (e.g., form feed, carriage return, and tab) would also have to be escaped
- Backslash (\) – The backslash is the escape character so it must be escaped (\b)

2.4 XMLDSig and White Space

Generally, XML is not sensitive to unquoted white space. However, when using XMLDSig it is important to maintain canonical form. This requires preserving white space.

For reference, see <https://www.w3.org/TR/xmlsig-core1/> and for more specifics <https://www.w3.org/TR/xmlsig-core1/#sec-c14nAlg>

3 TRANSLATING XML TO JSON

Translation from XML to JSON requires a few areas of special attention

3.1 Encoding Rules

The following rules are applied when generating JSON encoding of MDDF objects.

The JSON examples are derived from the following XML

```
<manifest:MediaManifest
  xmlns:manifest="http://www.movielabs.com/schema/manifest/v1.5/manifest"
  xmlns:md="http://www.movielabs.com/schema/md/v2.4/md"
  xsi:schemaLocation="http://www.movielabs.com/schema/manifest/v1.5/manifest
manifest-v1.5.xsd"
  ManifestID="SofaSpud.Example.trailers">
<manifest:Compatibility>
...
</manifest:MediaManifest>
```

- For top-level elements (e.g., “MediaManifest”), there are two conditions that would dictate the JSON structure
 - When the JSON object is used as a standalone document (e.g., stored as an asset in the cloud), the top-level object is retained. The structure of the overall document is the same in JSON and XML.

```
{
  "MediaManifest": {
    "_xmlns:manifest": "http://www.movielabs.com/schema/manifest/v1.8/manifest",
    "_xmlns:md": "http://www.movielabs.com/schema/md/v2.7/md",
    "_schemaLocation": "http://www.movielabs.com/schema/manifest/v1.8/manifest
manifest-v1.8.1.xsd",
    "_ManifestID": "SofaSpud.Example",
    "_updateNum": "1",
    "Compatibility": {
      "SpecVersion": "1.5",
      "Profile": "MMC-1"
    }
  },
}
```

- When the JSON object will be used as part of an API, and the context is understood (e.g., a query is made for a MediaManifest object), the top-level element is removed. For example,

```
{
  "Compatibility": {
    "SpecVersion": "1.5",
    "Profile": "MMC-1"
  },
  ...
}
```

- XML tag names are used as JSON keys:
 - Capitalization is preserved
 - Attributes are prefixed with an underscore (e.g., attribute “lang” becomes “_lang”)

- If an XML tag has no attributes, then the XML payload is the JSON value. For example,

```
<ResourceName>So You're Raising a Warrior</ResourceName>1
```

becomes

```
"ResourceName": "So You're Raising a Warrior"
```

- If an XML tag has attributes, then the XML payload is presented as the value of a sub-key labeled "value" followed by the attributes each as its own sub-key. For example:

```
<ResourceName lang="en">So You're Raising a Warrior</ResourceName>
```

becomes

```
"ResourceName":  
  {  
    "value": "So You're Raising a Warrior",  
    "_lang": "en",  
  }
```

- Any XML tag that has a cardinality > 1 is always presented as a JSON array, even when it only appears once. For example, assuming AlternateResourceName has cardinality 0..n,

```
<AlternateResourceName lang="zh-HK">Cengjing You  
Yongshi</AlternateResourceName>
```

becomes

```
"AlternateResourceName": [  
  {  
    "value": "Cengjing You Yongshi",  
    "_lang": "zh-HK",  
  }  
]
```

Note that these are the same rules used by EIDR (eidr.org). Thank you EIDR.

3.2 Translation Rules

The encoding rules assumes that JSON is being generated directly. However, if JSON is generated by transcoding from XML one must apply the same encoding rules applied in the previous section. For example, quotes in strings must be escaped.

3.3 Examples

The following MMC Example:

https://www.movelabs.com/md/mmc/examples/movies1/2019/MMC_Example1_simple.xml

in JSON:

https://www.movelabs.com/md/mmc/examples/movies1/2019/MMC_Example1_simple.json

¹ These examples are from EIDR.