

Common Catalog Format

IETF #120, Vancouver, 20-26 July 2024

Suhas Nandakumar, Will Law, Mo Zanaty

The catalog spec repo <https://github.com/moq-wg/catalog-format>

I-D available at <https://datatracker.ietf.org/doc/draft-ietf-moq-catalogformat/01/>

Agenda for this session

- Adoption of the Internet Draft
- Updates and changes since IETF #119 (March 16)
 - PR #36 - Removes obsolete sequence parameter from examples.
 - PR #39 - Adds an IANA registry for catalog fields
 - PR #40 - Adds in a new field to indicate if delta updates (patches) will be used by the publisher.
 - PR #42 - Add a new root object to hold inheritable track fields.
 - PR #47 - Renaming file post-workgroup adoption.
 - PR #49 - Reformatting to 80 chars + editorial fixes
- Current PR discussion
 - PR #64: Change 'packaging' to 'format'
 - PR #63: Adding type attribute for tracks
- Current issue discussion
 - Issue #53: commonTrackFields is a premature optimization
 - Issue #46: Add a field to allow suggested relative track prioritization
 - Issue #44: Streaming format should be a string not an integer

Notification of Adoption

Following the successful call-for-adoption on Feb 19, chairs issued **notification of adoption** on **March 4th**.

PR #47 renamed document from `draft-wilaw-moq-catalogformat-latest` to `draft-ietf-moq-catalogformat-latest`

New I-D draft available at

<https://datatracker.ietf.org/doc/draft-ietf-moq-catalogformat/01/>

Merged PR #39 - Adds an IANA registry for catalog fields

Purpose is to allow new fields to be added in the future without having to revise the RFC.

This document creates a new IANA registry for the Common Catalog fields. The registry is called "**MoQ Common Catalog Fields**".

This registry is managed by the IANA according to the *Specification Required* policy of [RFC5226]. The initial entries in the registry are:

5.2. Common Catalog Field Registry

This document creates a new IANA registry for the Common Catalog fields. The registry is called "MoQ Common Catalog Fields". This registry is managed by the IANA according to the Specification Required policy of [RFC5226]. The initial entries in the registry are:

Descriptive Name: Catalog version
Field Name: version
Required: yes
Location: R
JSON Type: String
Specification: [MoQCatalog]

Descriptive Name: Streaming format
Field Name: streamingFormat
Required: yes
Location: RC
JSON Type: String
Specification: [MoQCatalog]

Descriptive Name: Streaming format version
Field Name: streamingFormatVersion
Required: yes
Location: RC
JSON Type: String
Specification: [MoQCatalog]

...

Merged PR #40 - Add a new field to indicate if delta updates are to be expected

Purpose is to inform a player if it is necessary to support the delta updates (patch) mechanism in order to understand future catalog updates. This field SHOULD be omitted if it is FALSE.

Field	Name	Required	Location	JSON type
Supports delta updates	supportsDeltaUpdates	opt	RC	Boolean

```
{
  "version": 1,
  "sequence": 0,
  "streamingFormat": 1,
  "streamingFormatVersion": "0.2",
  "supportsDeltaUpdates": true,
  "renderGroup": 1,
  "packaging": "loc",
  "tracks": [
  ...
```

```
{
  "version": 1,
  "sequence": 0,
  "catalogs": [
    {
      "name": "catalog-for-format-one",
      "namespace": "sports.example.com/games/08-08-23/live",
      "streamingFormat": 1,
      "streamingFormatVersion": "0.2",
      "streamingFormatVersion": "0.2",
      "supportsDeltaUpdates": true,
    },
    {
      "name": "catalog-for-format-five",
      "namespace": "chat.example.com/games/08-08-23/chat",
      "streamingFormat": 5,
      "streamingFormatVersion": "1.6.2"
    }
  ]
}
```

Merged PR #42: Add a new root object to hold inheritable track fields.

- PR adds a new root object called **"commonTrackProperties"** to explicitly hold fields intended to be inherited by all tracks.
- These common objects will usually be K/V fields, however they may also be more complex custom-defined objects.
- All examples were also updated.

```
{
  "version": 1,
  "streamingFormat": 1,
  "streamingFormatVersion": "0.2",
  "commonTrackFields": {
    "namespace": "example.com/12345",
    "packaging": "cmf",
    "renderGroup": 1,
    "altGroup": 1
  },
  "tracks": [
    {
      "name": "video_4k",
      "selectionParams": {
        "codec": "avc1.640033",
        "mimeType": "video/mp4",
        "width": 3840,
        "height": 2160,
        "framerate": 30,
        "bitrate": 14931538
      }
    }
  ]
}
```

PR [#64](#): Change 'packaging' to 'format'

- Current spec defines a table of predefined values ('cmaf', 'loc')
- These are media specific and limiting - they don't support game play or other data-centric use-cases.
- Additionally, there are more attributes to a track format than just packaging.
- This PR replaces 'packaging' with 'format' and allows the application specification to define the allowed values.
- Question - should these be IANA registered, or can they be at the discretion of the application specification?

```
...
"tracks":[
  {
    "name": "game-instructions",
    "type": "datachannel",
    "format": "CBOR-special"
  },
  {
    "name": "media-timeline",
    "type": "timeline",
    "format": "csv"
  },
  {
    "name": "hd",
    "selectionParams": {"codec":"av01",...,"framerate":30},
    "altGroup":1,
    "renderGroup":1,
    "format": "cmaf"
  },
  {
    "name": "audio",
    "selectionParams":{"codec":"opus",...,"bitrate":32000},
    "packaging": "loc"
    "renderGroup":1,
    "format": "loc"
  }
]
}
```

PR [#63](#): Adding type attribute for tracks

- There are a number of use-cases in which it is necessary for the streaming application to be able to declare the 'type' of a track.
- For example, a 'data channel', or a 'timeline' track.
- The attribute can be a String.
- This field is optional.
- The application using the catalog format would define the contents of this field.
- Question - do we want to set a max length?

```
...
"tracks": [
  {
    "name": "game-instructions",
    "type": "datachannel",
    "format": "CBOR-special"
  },
  {
    "name": "live-timeline",
    "type": "timeline",
    "depends":["video","audio"]
  },
  {
    "name": "video",
    "type": "primary-video",
    "selectionParams":{"codec":"av01.0.08M.10.0.110.09","width":1920,
    "height":1080,"framerate":30,"bitrate":1500000}
  },
  {
    "name": "audio",
    "type": "primary-audio",
    "selectionParams":{"codec":"opus","samplerate":48000,"channelConfig":"2",
    "bitrate":32000}
  }
]
```


Issue #46: Add a field to allow suggested relative track prioritization

- Use case is a client subscribing to 3 tracks concurrently (say, audio, video and slides) but wanting to request the relay, under congestion conditions, to send the audio with higher priority than the slides and the slides with higher priority than the video.
- Simple numerical value, with lower numbers equalling higher priority
- Catalogs SHOULD indicate a relativeTrackPriority for all tracks if they supply it for any track, otherwise the client is left in an ambiguous state.
- Tracks MAY be assigned equal priority.
- Absence of a relativeTrackPriority among all tracks SHOULD be interpreted to all tracks having an equal priority.

```
{
  "version": 1,
  "streamingFormat": 1,
  "streamingFormatVersion": "0.2",
  "supportsDeltaUpdates": true,
  "commonTrackFields": {
    "renderGroup": 1,
    "packaging": "loc"
  },
  "tracks": [
    {
      "name": "video-hd",
      "selectionParams": {"codec": "av01", "width": 1920, "height": 1080, "bitrate": 5000000, "framerate": 30},
      "altGroup": 1,
      "relativeTrackPriority": 4
    },
    {
      "name": "video-sd",
      "selectionParams": {"codec": "av01", "width": 720, "height": 640, "bitrate": 3000000, "framerate": 30},
      "altGroup": 1,
      "relativeTrackPriority": 3
    },
    {
      "name": "slides",
      "selectionParams": {"codec": "av01", "width": 1920, "height": 1080, "bitrate": 1000000, "framerate": 15},
      "relativeTrackPriority": 2
    },
    {
      "name": "audio",
      "selectionParams": {"codec": "opus", "sampleRate": 48000, "channelConfig": "2", "bitrate": 32000},
      "relativeTrackPriority": 1
    }
  ]
}
```

Issue #53: commonTrackFields is a premature optimization

- Suggestion from Luke that commonTrackFields only purpose is to reduce catalog size.
- If we really want to do that, we should shift from JSON to binary
- Therefore let's remove commonTrackFields and simply repeat elements. This makes for a simpler parser at the expense of a more verbose catalog.

```
{
  "version": 1,
  "streamingFormat": 1,
  "streamingFormatVersion": "0.2",
  "commonTrackFields": {
    "namespace": "conference.example.com/conference123/alice",
    "packaging": "loc",
    "renderGroup": 1
  },
  "tracks": [
    {
      "name": "video",
      "selectionParams": {"codec": "av01.0.08M.10.0.110.09", ... "bitrate": 1500000}
    },
    {
      "name": "audio",
      "selectionParams": {"codec": "opus", ... "bitrate": 32000}
    }
  ]
}
```

Current

```
{
  "version": 1,
  "streamingFormat": 1,
  "streamingFormatVersion": "0.2",
  "tracks": [
    {
      "name": "video",
      "namespace": "conference.example.com/conference123/alice",
      "packaging": "loc",
      "renderGroup": 1
      "selectionParams": {"codec": "av01.0.08M.10.0.110.09", ... "bitrate": 1500000}
    },
    {
      "name": "audio",
      "namespace": "conference.example.com/conference123/alice",
      "packaging": "loc",
      "renderGroup": 1
      "selectionParams": {"codec": "opus", ... "bitrate": 32000}
    }
  ]
}
```

Proposed

Issue #44: Streaming format should be a string not an integer

- Comments from @JonathanLennox & @scottlamb
- Currently [1] streaming format is an integer, defined in an IANA table
- Proposal [2] is to convert it to a string. Suggestion to limit to ASCII [A-Za-z0-9_.-]
- I'd prefer to keep it UTF-8 to support non-Western character sets.
- Proposal [3]- coalesce StreamignFormat and StreamingFormatVersion fields into one.
- Proposal [4] - remove catalog version field. The Streaming format can define a catalog version as part of its versioning scheme.

```
{  
  "version": 1,  
  "streamingFormat": 1,  
  "streamingFormatVersion": "0.2",  
  ...  
}
```

1

```
{  
  "version": 1,  
  "streamingFormat": "WARP"  
  "streamingFormatVersion": "0.2",  
  ...  
}
```

2

```
{  
  "version": 1,  
  "streamingFormat": "WARPv0.2"  
  ...  
}
```

3

```
{  
  "streamingFormat": "变形0.2"  
  ...  
}
```

4