



MSF & CMSF Updates & Issues

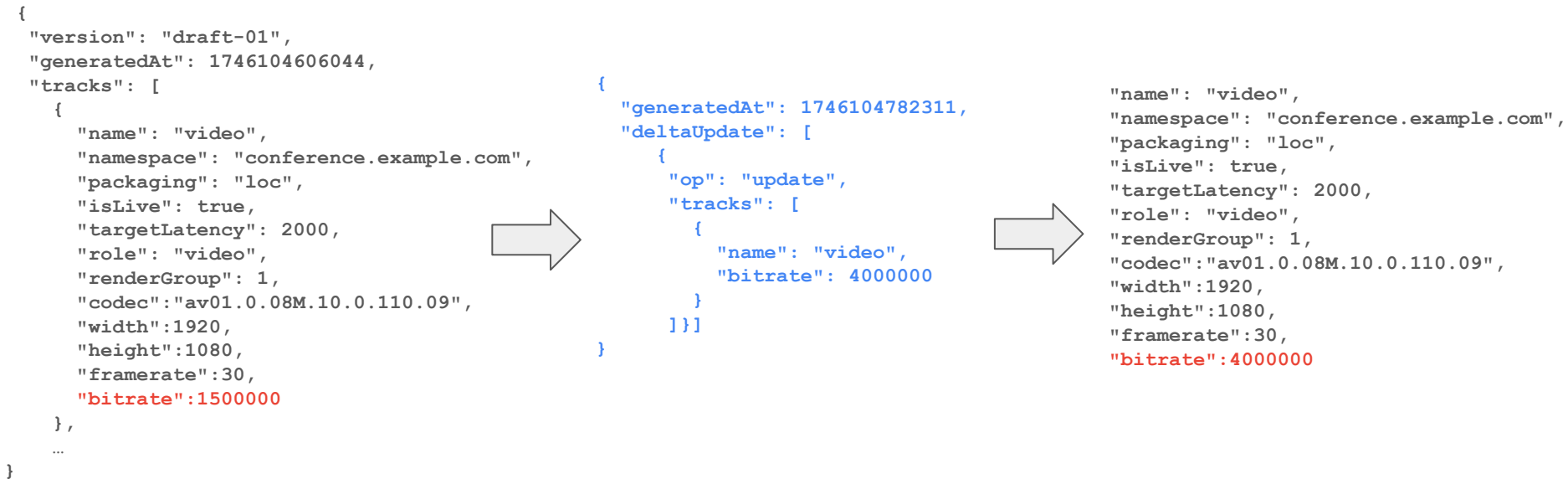
<https://github.com/moq-wg/msf>

Will Law, Suhas Nandakumar, Gwendal Simon

IETF #126, Vienna July 2026

Merged (#187): Updating select properties of a track via delta update

New "update" action added to delta updates - in addition to "add" and "remove"



Merged (#186): New Track and Object properties to carry init data

There have been requests to allow init properties (such as width/height) to vary during the life of track.

MSF_INITIALIZATION Track property type: 0x79

Carries the initialization data for the track. This initialization data is immutable over the life of the track.

If used, publishers MUST include the MSF_INITIALIZATION track property in the PUBLISH message (publisher-initiated flow) or SUBSCRIBE_OK message (subscriber-initiated flow).

MSF_INITIALIZATION Object property type: 0x79

Carries initialization data for the track, when that initialization data might change over the life of the track. Since subscription might occur at any time, this property needs to be repeated. Publishers SHOULD add this property to the first Object in each Group.

```
"initDataList": [  
  {  
    "id": "init1",  
    "type": "inline",  
    "data": "AAAAHGZ0e..."  
  },  
  {  
    "id": "init2",  
    "type": "inline",  
    "data": "AAAAHGZ0e..."  
  },  
  {  
    "id": "init3",  
    "type": "track-property"  
  },  
  {  
    "id": "init4",  
    "type": "object-property"  
  }  
]
```

1

Should MSF use Track and Object properties in the reserved application space or register them in the MOQT table?

MOQT 5.8. Properties

- MOQ Properties - we wish to define the following registration policies:
 - 0x00 to 0x77: Standards Action or IESG Approval (1-byte encoding)
 - 0x78 to 0x7F: Reserved for application-specific use (1-byte encoding, no registration permitted)
 - 0x80 to 0x37FF: Specification Required (2-byte encoding)
 - 0x3800 to 0x3FFF: Reserved for application-specific use (2-byte encoding, no registration permitted)
 - 0x4000 to 0x7FFF: Reserved for Mandatory Track Properties (see [Section 2.5.1](#)). Properties registered in this range **MUST** have Track scope; Object scope properties **MUST NOT** be registered in this range.
 - 0x8000 and above: First Come First Served

Should we be here?

We are here now.

Code points reserved for application-specific use will never be allocated by IANA. Applications using these values do not need to coordinate with IANA.

PR#122: Switching across tracks with different Group/GOP lengths



- Catalog signals track **keyframeSpacing**: A=2, B=4, C=8.
- All tracks MUST start with same group ID. Does not need to be 0.
- Group/GOP duration MUST be constant over the life of the track.

A to B	A to C	B to A	B to C	C to A	C to B
0	0	0	0	0	0
0.25	0.125	0.5	0.125	0.5	0.25
0.5	0.25	1	0.25	1	0.5
0.75	0.375	1.5	0.375	1.5	0.75
1	0.5	2	0.5	2	1
1.25	0.625	2.5	0.625	2.5	1.25
1.5	0.75	3	0.75	3	1.5
1.75	0.875	3.5	0.875	3.5	1.75
2	1	4	1	4	2
2.25	1.125	4.5	1.125	4.5	2.25
2.5	1.25	5	1.25	5	2.5
2.75	1.375	5.5	1.375	5.5	2.75
3	1.5	6	1.5	6	3
3.25	1.625	6.5	1.625	6.5	3.25
3.5	1.75	7	1.75	7	3.5
3.75	1.875	7.5	1.875	7.5	3.75
4	2	8	2	8	4
4.25	2.125	8.5	2.125	8.5	4.25
4.5	2.25	9	2.25	9	4.5
4.75	2.375	9.5	2.375	9.5	4.75
5	2.5	10	2.5	10	5
5.25	2.625	10.5	2.625	10.5	5.25
5.5	2.75	11	2.75	11	5.5
5.75	2.875	11.5	2.875	11.5	5.75

1. Player joins track with lowest keyframe spacing (A). This gives it the fastest join at the lowest latency.
2. Given a source track with keyframeSpacing S and a target track with keyframeSpacing T, a switch at Group G and Object N of the source track is possible if $(G \cdot S + N) / T$ is an integer. The target group ID will be that integer.

Issue 130: Replacing catalog with 'nothing'

Let me explain.

MOQT allows empty track names. If we define the catalog to be the null or empty track name by default, then we don't need to specify its name. The URL syntax collapses from

```
moqt://example.com/relay-app#msf:app-broadcastID--catalog
```

to

```
moqt://example.com/relay-app#msf:app-broadcastID
```

We only need to communicate the namespace, and if the type prefix is msf: then a player knows it can retrieve the catalog by requesting the null track. This allows as many other custom catalogs as you want

```
moqt://example.com/relay-app#msf:app-broadcastID // default catalog
```

```
moqt://example.com/relay-app#msf:app-broadcastID--free
```

```
moqt://example.com/relay-app#msf:app-broadcastID--paid
```

PR #190: Add Event Timeline track Property to carry header data

- Event timeline definition is just an array of records.
- In practice, much of this data could benefit from a static header - for initialization, metadata, context etc.
- This PR define a track property 0x7B for a media timeline track, which holds application-defined header data for that track.
- This track property is optional. All clients MUST check for the presence of this property before processing the track payload.

```
[  
  { "I": [0,0], "data": [47.1812, 8.4592] },  
  { "I": [1,0], "data": [47.1662, 8.5155] },  
  { "I": [2,0], "data": [47.1534, 8.4871] },  
  { "I": [3,0], "data": [47.1723, 8.5312] },  
  { "I": [4,0], "data": [47.1489, 8.5034] },  
  { "I": [5,0], "data": [47.1601, 8.4743] },  
  { "I": [6,0], "data": [47.1845, 8.5221] },  
  { "I": [7,0], "data": [47.1378, 8.4965] },  
  { "I": [8,0], "data": [47.1956, 8.5087] },  
]
```

Track property 0x7B carries this JSON payload

```
{  
  "Customer" : "Crop Inspection Services"  
  "Drone ID" : "7h3fg23"  
  "Job Site" : "Big Fruit Farm"  
  "Billing code" : 5300  
  "Date" : "23.09.26"  
}
```

PR #169 - Update MOQT mapping details in media transmission section

The MOQT Groups and MOQT Objects need to be mapped to MOQT Streams. This mapping is defined by the packaging type of each track.

Irrespective of the packaging type in place, each MOQT Object MUST be placed on MOQT sub-group 0.

1. Is this correct? What about a future Temporal Scalability packaging mode?
2. Do we relax to a "SHOULD be placed on sub-group 0"?
3. Or do we modify in the future IF Temporal Scalability is introduced?

Issue #188: Static vs dynamic catalogs

Could there be a way to signal when no more tracks will be added/removed/modified?

Currently we have the **isComplete** flag, which signals that no new tracks will be added, however it also signals that no new content will be added, since its purpose is to indicate a transition from live to VOD.

Possible solutions

1. Relax the meaning of this flag to remove the requirement that no new content (within the existing tracks) is published. Within each track, we have the **isLive** flag to signal if that track will be adding new content.
2. Or we create a new flag to signal no new tracks (but tracks continue to produce new content) , and a second to signal no new content . We then deprecate **isComplete**.

Thank you for your time.