Use timed metadata with an ultra low latency stream target in Wowza Streaming Cloud

Originally Published on 09/06/2018 | Updated on 03/26/2019 11:52 am PDT

Timed metadata can add interactivity to ultra low latency streams and provide engaging content experiences for your viewers. The Wowza Streaming Cloud™ service with Ultra Low Latency leverages both AMF and ID3 metadata formats as streams move from encoder to player.

In this article, learn how to ingest AMF data into ultra low latency streams in Wowza Streaming Cloud and consider available playback options for timed metadata.

About timed metadata and ultra low latency streaming

Ultra low latency origin servers can ingest AMF data in source streams over RTMP or WOWZ. From there, the origin servers send the AMF data to ultra low latency stream targets for playback using WOWZ over WebSockets. Meanwhile, if HLS is enabled as a fallback for WOWZ ultra low latency streams, Wowza Streaming Cloud converts incoming AMF data into ID3 tags for playback in HLS.

Ingest AMF metadata into a Wowza Streaming Cloud ultra low latency stream target

Wowza Streaming Cloud ultra low latency stream targets can receive AMF metadata with any single-bitrate source stream received over RTMP or WOWZ from an external, third-party encoder, as well as from Wowza Streaming Engine™ and from Wowza GoCoder™ SDK broadcast apps.

To use Wowza Streaming Engine to send a stream with AMF data directly to an ultra low latency stream target, first insert the timed metadata using Wowza Streaming Engine. Next, ingest the stream from Wowza Streaming Engine into an ultra low latency stream target in Wowza Streaming Cloud. For instructions on these workflows, see More resources.

With Wowza GoCoder SDK broadcast apps, first insert AMF data programmatically, then send the stream with AMF data directly to an ultra low latency stream target. For instructions on
these workflows, see More resources.

For additional instructions on sending AMF metadata with a third-party encoder, see the manufacturer’s documentation.

**Note:** In order for Wowza Streaming Cloud to convert AMF metadata to ID3 tags for HLS playback, it must be in the following format:

- The AMF data must be included in in a top-level AMF data object (AMFDataObj). Within that AMF data object, you can include key:value pairs that include AMF data of additional types, such as string, Boolean, list, or even a nested object.
- The AMFDataObj object must include two properties: (1) a key called payload whose value is a string of data to be converted and (2) a key called wowzaConverter whose value is basic_string. For a code example showing how to format incoming AMF data, see Inject timed metadata using a Wowza Streaming Engine HTTP provider.

**Verify the metadata in the stream**

To verify that the AMF data is showing up in an ultra low latency stream and to verify that AMF data converted to ID3 tags is showing up in an HLS fallback stream, view the console log of a self-hosted page with a Wowza Player embed code.

**Verify AMF data in an ultra low latency WOWZ over WebSockets stream**

1. Set up an ultra low latency stream in Wowza Streaming Cloud.
2. Set up a source that sends AMF data.
3. Using Wowza Player for Ultra Low Latency, set up playback for the ultra low latency stream in a self-hosted webpage. Include this code as part of the Wowza Player embed code:
4. Start the source for the ultra low latency stream.
5. Open the self-hosted page with Wowza Player, play the stream, and view the console log to verify that metadata shows up in the stream.
6. Stop the source.

Verify AMF data converted to ID3 tags in an HLS fallback stream

1. Set up an ultra low latency stream with HLS fallback enabled in Wowza Streaming Cloud.
2. Set up a source that sends AMF data.
3. Using Wowza Player for Ultra Low Latency, set up playback for the fallback HLS stream in a self-hosted webpage. Include this code as part of the Wowza Player embed code:

```javascript
myPlayer = WowzaPlayer.create("playerElement", {
  license: "XXXXX-XXXXX-XXXXX-XXXXX-XXXXX-XXXXX",
  sourceURL: "wss://edge.cdn.wowza.com/live/_definst_/[stream_name]/stream.ws"
});
myPlayer.onMetadata(function (event) {
  var now = new Date();
  var seconds = now.getTime();
  var diff = seconds - event.value.injectTime;
  console.log("INFO: " + seconds + " onMetaData event:" + event.type + " delay:" + diff);
  console.log(event);
});
```

```javascript
myPlayer = WowzaPlayer.create("playerElement", {
  license: "XXXXX-XXXXX-XXXXX-XXXXX-XXXXX-XXXXX",
  sourceURL: "https://[wowzasubdomain]-i.akamaihd.net/hls/live/[provider_stream_id]/[stream_name]/playlist.m3u8"
});
myPlayer.onMetadata(function (event) {
  var now = new Date();
  var seconds = now.getTime();
  var diff = seconds - event.value.injectTime;
  console.log("INFO: " + seconds + " onMetaData event:" + event.type + " delay:" + diff);
  console.log(event);
});
```

4. Start the source for the ultra low latency stream.
5. Open the self-hosted page with Wowza Player, play the stream, and view the console log to verify that metadata shows up in the stream.
Playback options

After configuring and testing a workflow where AMF data flows into a Wowza Streaming Cloud ultra low latency stream target, you can consider playback options for timed metadata in your ultra low latency stream and its backup HLS stream.

Ultra low latency stream with metadata

You can use the AMF data in ultra low latency streams sent over a WebSocket connection to trigger events during playback. Use Wowza Player for browser-based playback or Wowza GoCoder SDK for mobile app playback to provide ultra low latency stream interactivity using AMF data. See More resources for more information on these workflows.

Backup HLS stream with metadata

You may want your ultra low latency stream target to have a backup HLS stream, which provides a fallback connection in the event that WebSocket connection attempts fail and provides an alternative connection if your audience exceeds the maximum simultaneous viewer limit for an ultra low latency stream target. In order to pass AMF data events to a backup HLS stream, Wowza Streaming Cloud automatically converts AMF data to ID3 tags, a metadata format that HLS supports.

Note: Wowza Streaming Cloud HLS streams that flow through stream target types other than ultra low latency must enable the convertAMFData stream target property to convert AMF data to ID3 tags. See Ingest and convert timed metadata with the Wowza Streaming Cloud REST API for more information.

In order to provide interactive playback experiences on backup HLS streams, you can leverage Wowza™ Player to listen for ID3 tags and trigger events during playback. See More resources for information on this workflow.

At this time, the GoCoder SDK doesn’t pass ID3 tags to native Apple and Android player frameworks, but you can build an HLS player independently that will leverage the timed metadata to invoke an action during playback.
More resources

- Inject timed metadata using a Wowza Streaming Engine HTTP provider
- Send a stream from Wowza Streaming Engine to an ultra low latency stream target in Wowza Streaming Cloud using Wowza APIs
- Send and receive events with Wowza GoCoder SDK for iOS
- Send and receive events with Wowza GoCoder SDK for Android
- Use timed metadata in HLS streams with the Wowza Streaming Cloud REST API
- Listen for metadata and trigger an action with the Wowza Player JavaScript API
- Using timed metadata with Wowza GoCoder SDK