

StreamhubAnalytics plugin for Roku devices

8th of Jan. 2017	1.0	Basic tracking, QoS and Ads
23th of Nov. 2017	1.1	Frontloading of metadata
8th of Dec. 2017	1.2	Added requirement for onMediaLoaded event.

In order to track videos, qos, metadata and ads engagement for you Roku application, we provide a plugin in the form of a Brightscript file which you can drop in your application and use from your application client code.

We provide a sample Roku application that you can browse around in order to familiarise yourself with the integrations steps.

1 Dowload the plugin sample application

You can dowload the Roku sample app and plugin at

<http://streamhub-static-content.s3.amazonaws.com/plugins/genericplugins/roku-plugin.zip>

2 Add the plugin file to your application source folder

Locate the file **StreamhubAnalyticsRokuBridge.brs** in the application source code and add it to your own application

3 Setup the SetPositionNotificationPeriod parameter

In your application code, upon your videoplayer instance object, call `SetPositionNotificationPeriod(1)` to indicate to your videoplayer that you want to be notified of playback progression every seconds.

4 Modify your application event loop to intercepts video player message events

In your **main.brs** file, add the following message handlers to your event loop to capture events of interests and forward them to the StreamhubAnalytics plugin for tracking:

- isStreamStarted() notifies of any new video stream being started.
- isPlaybackPosition() notifies of in-progress playback
- isFullResult() notifies about stream playback completes

Initialise the plugin in the isStreamStarted() message handler

#1 Create a new instance of the StreamhubAnalytics plugin and initialise it

The call to the StreamhubAnalytics function will return an Object that will be used to perform subsequent calls.

Brightscript code

```
m.streamhub = StreamhubAnalytics(partnerId as String, endPoint as String, playerId as String, isLive as Boolean, userId as String, analyticsId as String)
```

Property name	Required	Short description	Remarks
partnerId	X	identifier of the OVP partner. Provided to you by your Streamhub account manager	<i>Provided to you by your Streamhub account manager.</i>
endPoint	X	[http/https]://stats.streamhub.io is the production endPoint for submitting stats. However, you could choose to use your own mockup for development purpose.	
analyticsId	X	A unique identifier in our system for each combination of a content owner and a partner.	<i>Provided to you by your Streamhub account manager.</i>
playerId	X	A unique identifier for a video player across your ecosystem of websites and application.	<i>Has to be generated by us or communicated to us. Important: player ids not communicated to us, will result in unprocessed views (ticks will be stored and can be processed at a later time though)</i>
isLive	X	indicates if the media currently played is a Vod content or a Live event	true false
userId		userId token for subscribed video service (SVOD)	

#2 Call onMediaLoaded

Right after having instantiated the StreamhubAnalytics instance, call the *onMediaLoaded* instance method to signal the plugin that the app is starting to request a video file.

Brightscript code

```
m.streamhub.onMediaLoaded( m.streamhub, 0 )
```

#3 Provide the media duration

This call is necessary for:

- Precomputing completion rates stop points at (1%, 25%, 50%, 75%, 95%)
- Communicating the program duration as a metadata

Brightscript code

```
m.streamhub.setDuration( m.streamhub, m.player.GetPlaybackDuration() )
```

#4 Provide the media identifier that will be used to process stats for this media

Brightscript code

```
m.streamhub.onMediaReady( m.streamhub, "Some-Video-Identifier" )
```

#5 Provide Video Metadata

Call `addMediaMetadata(config as Object, key as String, value as String)` on the streamhub instance for each metadata.

Currently, we support only 4 types of metadata:

Property (as String)	Example values	Additional notes
title	Jeff Han Demos his Breakthrough Touchscreen	String, the program title
categories	TED Conference Technology"	String, multiples categories must be separated via a pipe (" ") character
duration	Provided via → Function StreamhubAnalyticsSetDuration(config as Object, pDuration as Integer)	You don't really call <code>m.streamhub.addMediaMetadata(m.streamhub, "duration", ...)</code> because the duration is already provided in #2 Provide the media duration
playerTitle	Airbeem Roku Player	String, a user friendly name to identify this roku app player in your Streamhub Dashboard.

Brightscript code

'Provide metadata infos

```
m.streamhub.addMediaMetadata( m.streamhub, "title", "Jeff Han Demos his Breakthrough Touchscreen" )  
m.streamhub.addMediaMetadata( m.streamhub, "playerTitle", "Airbeam Roku Player" )  
m.streamhub.addMediaMetadata( m.streamhub, "categories", "TED|Conference|Technology" )
```

'Make the actual request to our log collector

```
m.streamhub.onMediaMetadata( m.streamhub )
```

#6 Call the onMediaStart plugin method to indicate the stream is starting

Brightscript code

```
m.streamhub.onMediaStart( m.streamhub, 0 )
```

Track playback progress in the isPlaybackPosition() event handler

Call the onTick plugin method of the plugin to track playback viewing. You can retrieve the playback position paramter (e.g. *m.position* in the code example) via the `message.GetIndex()` function.

Brightscript code

```
m.position = msg.GetIndex()  
m.streamhub.onTick( m.streamhub, m.position )
```

Track playback completion in the isFullResult() event handler

Call the onMediaComplete method of the plugin to track playback completion

Brightscript code

```
m.streamhub.onMediaComplete( m.streamhub, m.position )
```

Track QoS buffering event via isSegmentDownloadStarted and isDownloadSegmentInfo message events

You can write your own logic to capture and measure buffering events that occur during playback by observing the `msg.isSegmentDownloadStarted()` and `msg.isDownloadSegmentInfo()` message handler in your application event loop.

<https://sdkdocs.roku.com/display/sdkdoc/roVideoPlayerEvent>

Call the `StreamhubAnalyticsOnMediaBufferedComplete` method to track those buffering sequences.

Brightscript code

```
m.streamhub.onMediaBufferedComplete(m.streamhub, m.position, bufferingTime)
```

Note: If that is possible, we like to make the distinction between the buffering that happens at loading time (before the playback actually starts) versus the rebuffering that happens during playback.

If you have the ability to capture the first buffering event, we recommend to call the `onMediaPrebuffering` method to track it.

Brightscript code

```
m.streamhub.onMediaPrebuffering(m.streamhub, m.position, bufferingTime)
```

Tracking Ads via Roku Advertising Framework (RAF)

Read the documentation about RAF to learn about Ad rendering and tracking

<https://sdkdocs.roku.com/display/sdkdoc/Integrating+the+Roku+Advertising+Framework>

The sample channels section contains sample applications integrating the framework

Ad structure → adId

<https://sdkdocs.roku.com/display/sdkdoc/Integrating+the+Roku+Advertising+Framework#IntegratingtheRokuAdvertisingFramework-AdStructure>

```
Ad Structure

adPods : [
  +viewed : Boolean,
  +renderSequence : "preroll" | "midroll" | "postroll",
  +duration : Float (in s),
  renderTime : Float (in s),
  backfilled : Boolean,
  +ads: [
    +duration : Float (in s),
    +streamFormat : String,
    +adServer : String (URL),
    adId : String,
    adTitle : String,
    advertiser : String,
    creativeId : String,
    clickThrough : String (URL),
    +streams : [
      +url : String (URL),
      +bitrate : Int (in kbps),
      +width : Int,
      +height : Int,
      +mimeType : String,
      provider : String
    ]
  ]
]
```

Ad Tracking Callback via `setTrackingCallback(callback as Function, obj as Object)`

<https://sdkdocs.roku.com/display/sdkdoc/Integrating+the+Roku+Advertising+Framework#IntegratingtheRokuAdvertisingFramework-FrequencyCappingandTargetingusingRIDA>

Tracking events

<https://sdkdocs.roku.com/display/sdkdoc/Integrating+the+Roku+Advertising+Framework#IntegratingtheRokuAdvertisingFramework-Tracking>

Report Ad event to Streamhub

We are interested in the 5 following events: Impression, FirstQuartile, Midpoint, ThirdQuartile and Complete, which we report as percentiles of ad viewing completion:

RAF event	Streamhub percentile
Impression	0
FirstQuartile	25
Midpoint	50
ThirdQuartile	75
Complete	100

```
m.streamhub.trackAd( m.streamhub, adId, percentile )
```

```
Example: m.streamhub.trackAd( m.streamhub, "56327153217", 0 )
```