

Brief Overview

This document overviews for Brightcove customers how they can configure and deploy the open source Adobe Analytics (Omniiture) plugin for the Brightcove Player (based on Video.js), both old 1.x and new 5.x player versions. The plugin was developed for integration with Adobe Heartbeat library 1.5.2 and Adobe AppMeasurement 1.4.1.

The Adobe AppMeasurement library is used for what's referred to as 'Milestone' tracking and requires values such as 'eVars', 'props', and 'event' numbers for set-up.

The newer Adobe Heartbeat plugin eliminates any need for milestones and the definition of other values such as eVars, props and events. Events are dispatched as needed throughout the video and advertisements, contain video/player metadata and are trafficked to the Adobe Analytics tracking servers. From here, customers can create their own processing rules to aggregate this data within their reports.

While it is possible to use both Adobe Heartbeat and Milestone at the same time with the plugin, customers should work with Adobe to get set up with one or the other tracking type. A simple flag exists to determine which integration to implement.

Adobe Heartbeat v1.5.2

What's Included

- Heartbeat tracking
 - Adobe libraries & APIs have been implemented with the Brightcove Player APIs & events to provide heartbeat metrics during content playback
- Ad Tracking
 - Adobe libraries & APIs have been implemented with the Brightcove Player APIs & events to provide heartbeat metrics during ad playback
- Quality of Service Events
 - Most QoS-supported metrics found out-of-box in Adobe libraries
 - Custom work with Brightcove API was performed to track bitrate, bitrate changes, buffer events, and startUpTime
 - bitrates and bitrate change events only supported for Brightcove Players which load HLS Video with Flash Tech
- Ability to disable each tracking type. e.g, Heartbeat tracking can be disabled simply by setting the main config Object's **Heartbeat.DISABLE : false**
- Channel definition tracks as top-level domain (e.g., "domain" would be tracked for <http://blog.domain.com/>)
- Reporting Suites and other Adobe account tracking information can be customized through plugin-level configuration
- Support for autoplay

What's Not Currently Included

- Chapter Event Tracking

- Custom Metadata Tracking
- Any additional custom tracking scenarios (e.g., defining specific tracking servers based on the domain currently loaded)

If specific enhancements or subject matter expertise is needed, Brightcove customers are welcome to engage with Brightcove's Global Services team.

Adobe AppMeasurement & Milestone Tracking (“old-world” tracking)

What's Included

- Out-of-box the plugin will implement tracking against any events the BC Player VideoJS API can dispatch.
- Context Data Mapping & Milestones tracking can be defined as documented within Adobe by using the `bc_data_mapping` object in the Configuration Object. See Example in Appendix I
- Additional Brightcove-supported VideoJS events are supported for tracking. For a complete list, see Appendix II
- The current volume (a decimal between 0-1) will be tracked to the 'bc_volumechange' eVar property
- Tracking of following Milestones:
 - 25%
 - 50%
 - 75%
 - Additional milestones require code changes at this time.

What's Not Included

- Tracking any additional events which are not included in list of Appendix II. These events would need to be added to the codebase through custom listeners/handlers and additional code.
- Tracking additional metadata that needs to be set as evars or props.
- Any additional custom tracking scenarios (e.g., defining specific tracking servers based on the domain currently loaded)
- **eVars/props** are NOT tracked out-of-box for the following VideoJS events:
 - `bc_ad_resumed`
 - `bc_fullscreen_enter`
 - `bc_fullscreen_exit`
 - `bc_social_opened`
 - `bc_social_closed`
- The following Ad events rely on a Google IMA SDK (`google.ima.AdEvent`) which does not appear to be present in the current code, and therefore will not dispatch.
 - `bc_ad_start`
 - `bc_ad_complete`
 - `bc_ad_resumed`
 - `bc_ad_click`
- Support for Autoplay; Plugin will potentially not be ready for tracking when Player is set to autoplay videos; thus tracking will not correctly appear as expected.

If specific enhancements or subject matter expertise is needed, Brightcove customers are welcome to engage with Brightcove's Global Services team.

Appendix I - Context Data Mapping & Milestone Set-up

```
bc_data_mapping : {  
  name : "eVar202,prop2", // eVars, props only  
  segment : "eVar203",  
  contentType : "eVar201",  
  timePlayed : "event203",  
  view : "event201", // ***VIDEO START***  
  segmentView : "event202",  
  complete : "event207", // ***VIDEO COMPLETE***  
  milestones : { /** PERCENTAGES CAN BE CUSTOMIZED **/  
    25 : "event204",  
    50 : "event205",  
    75 : "event206"  
  }  
}
```

Appendix II - Additional Brightcove-Supported Event Tracking

```
bc_volumechange : {  
  event : 'event208',  
  evar : 'eVar204'  
},  
bc_ad_pause : {  
  event : 'event209',  
  evar : 'eVar205' // currentTime  
},  
bc_ad_resumed : {  
  event : 'event210',  
  evar : 'eVar205' // currentTime  
},  
bc_ad_click : {  
  event : 'event211',  
  evar : 'eVar206,eVar207'  
},  
bc_fullscreen_enter : {  
  event : 'event212'  
},  
bc_fullscreen_exit : {  
  event : 'event213'  
},  
bc_social_opened : {  
  event : 'event214'  
},  
bc_social_closed : {
```

```
    event : 'event215'  
  }
```

Set-up Instructions

A Breakdown of the Plugin ZIP files

Plugin Files

The ZIP contains 3 folders:

- configuration
 - Contains a single configuration file example. See more in: **Configuring the Plugin**
- production-lib
 - bcc_adobe_analytics.js
 - A compiled + concatenated version of our plugin plus the 3 Adobe dependency files.
 - bcc_adobe_analytics.min.js
 - This is the same file as above, however, the code has been *minified* and the file size is smaller. It is recommended that customers use either one of these versions of the Plugin.
- Also included are example HTML files which display loading the plugin in-page.

Dependencies

The plugin has several dependencies:

- Adobe Dependencies:
 - AppMeasurement.js
 - VideoHeartbeat.min.js
 - VisitorAPI.js
- jQuery
 - jQuery is required for this plugin to run. If jQuery is not inherently loaded into the global scope, then it can be enabled through a plugin parameter. See: **Configuring The Plugin**

Implementing the Plugin for a Brightcove Player

The plugin (i.e., bcc_adobe_analytics.min.js) can be implemented for a player either **in-page** as shown in the example, or directly in a player through the Video Cloud studio using Plugin Settings in the Players module.

If using the latter, then the rest of the settings will look like this:

JS:

http://domain.com/path/to/bcc_adobe_analytics.min.js

CSS:

<<leave blank>>

Name: BCGSAdobeAnalyticsPlugin

Plugin Options: ***see **Configuring The Plugin**

Configuring The Plugin

The plugin accepts two configuration parameters:

"jQuery"	<p>String - <i>optional</i> - The version of jQuery that the plugin should load. Use only if jQuery is not being loaded within the global scope.</p> <p>e.g., jQuery: "2.1.4"</p>
"options"	<p>Object - <i>required</i> - A configuration object which defines all Adobe account settings, and if using Milestone tracking, also defines contextDataMapping.</p> <p>For an example, please see the configuration/config.js from the Deliverable ZIP. This contains the object with all possible configurable parameters, along with comments.</p> <p>You'll notice that this object is being placed into a var named bcgs_adobe_config. Rather than defining this entire JSON Object within the "objects" parameter, I am storing it in a global variable which is then referenced when kicking off the plugin on my example pages (index.html, uncompiled.html, no-jquery.html).</p> <p>If loading the Plugin onto the Player via the Studio Settings then global variables cannot be used to store the "options" Object. Instead, the entire JSON Object should be used as a value for the "options" parameter.</p>