**DwBP - 9 - Implementing the Overlay Plugin Using Code**

Before we start looking at code, it is appropriate to get a base knowledge of plugins. As the slide indicates, plugins are the way to enhance Brightcove Player.

We here at Brightcove follow this pattern. The base player is enhanced by plugins. Some are always loaded in into the player, like the plugin that displays and handles errors, and the plugin that plays HLS video. Brightcove supplies others, like a plugin to play IMA ads, that can be optionally loaded.

All the samples in the documentation roll the code into a plugin, if possible.

Loading plugins into the player can be done in a number of ways. Studio’s PLAYERS module allows provides a UI to load numerous plugins, like the Social, Endscreen and IMA advertising plugins.

Also provided is a UI to load plugins, either Brightcove supplied or custom plugins which you can write.

Since this course is about Brightcove Player development, in this video we will write code in the page to load and configure a plugin.

Now we have some idea about plugins, and there is loads of information in the documentation about plugins, let’s look this use case and the code to solve it.

The wanted behavior is to display an overlay from when the player starts to play the video, and then remove the overlay when the playhead reaches four seconds. Also, the overlay needs to display dynamic information from the mediainfo object, otherwise the configuration of the Overlay plugin could be done in Studio.

After starting the video the overlay immediately displays, showing the tags tigers and wildlife read from the tags property of the mediainfo object, then the overlay is removed at 4 seconds.

Know we know exactly what we want to have happen, let’s look at the code that makes it work.

When a plugin is used, generally both a CSS and JavaScript file are needed to make the plugin function properly. Here we see the CSS file associated with the Overlay plugin. This URL to the file can be found either in the Display Overlay Plugin doc, or the Plugin Version Reference.

We also need to add in the URL to the JavaScript code for the plugin.

Remember to use info from the mediainfo object, we must wait for the loadstart event to be dispatched.

In that event handler, the following is performed:

-Read the tags from the mediainfo object  
-Convert the data type of the tags from an array to a comma separated list  
-Call the overlay method, defined in the JavaScript file  
-Define the content that should be displayed in the overlay, using HTML and the tags from the mediainfo object  
-Define when the overlay should first be displayed. In this case we are using the player event start. Any player event, or a time in seconds, is a valid value.  
-Define when the overlay should be removed, in this case when the video playhead hits 4 seconds.  
-Define where the overlay should be displayed, in this case bottom left. There are 8 alignment options.

Let’s see the results again of the code. Yep, still working.

As a quick review, let’s look at a slide. This reinforces:

1. Use a script tag to include the JavaScript for the plugin
2. Use a link tag to include the CSS for the plugin
3. Call the method defined for the plugin’s functionality
4. When configuration code for the plugin is placed on the page, it only adds the plugin’s functionality to the player on the page. If the plugin is added in Studio, it will add the functionality to all instances of that player.
5. It is probably a best practice to add the plugin to the player in Studio if the use of the plugin will be in the same manner all the time. If the plugin needs to be altered by changing options and/or behaviors of the plugin, that would be a reason to place it on the HTML page.

That concludes the information about plugins and implementing them in code.

Note that there are numerous player development docs about plugins, as well as more about using plugins in Studio.

Hopefully this video has been helpful. Thanks for your time and being a Brightcove customer!!