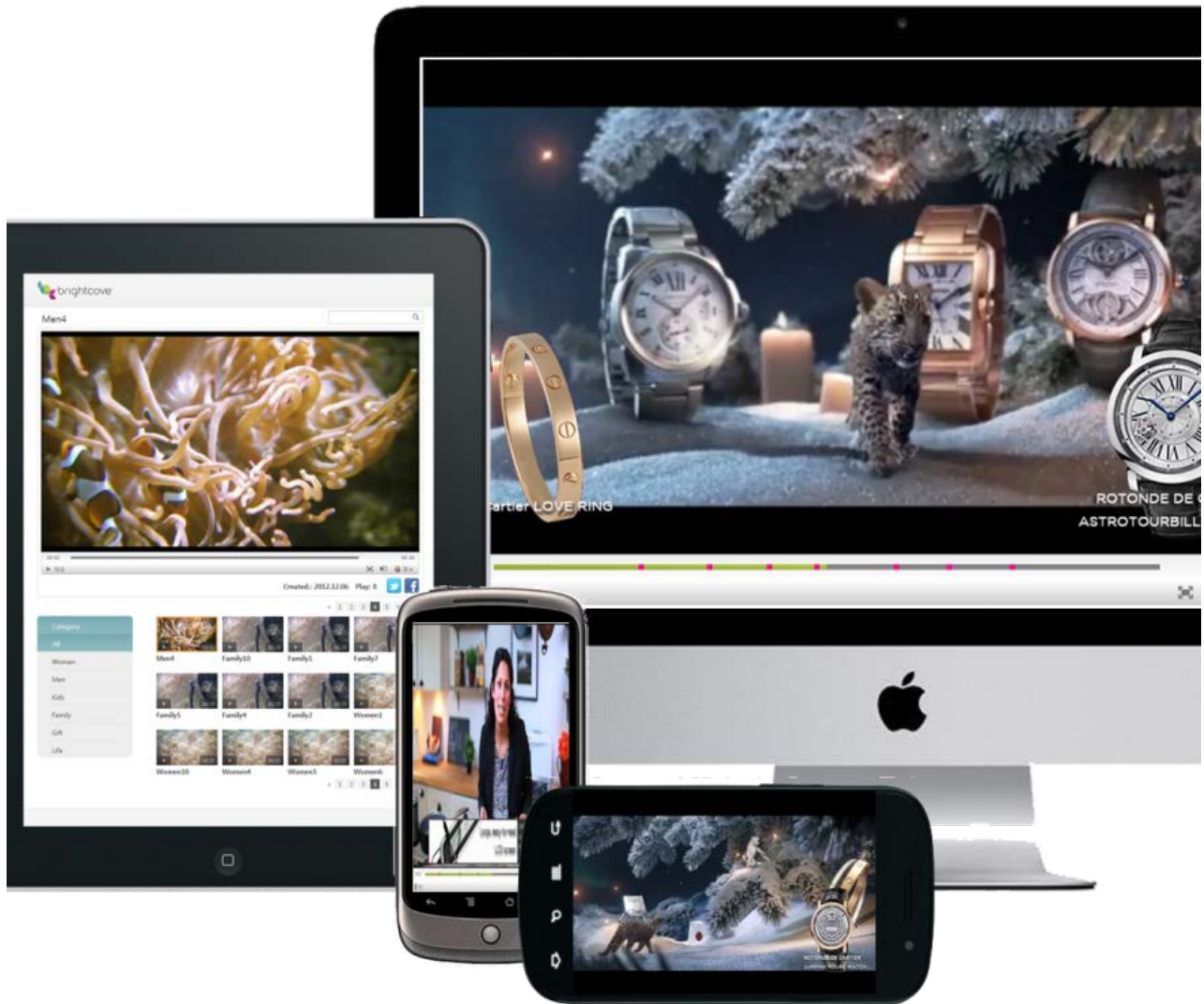


Brightcove V-Commerce Template User's Guide



Last updated: May 13, 2013

What is Brightcove V-Commerce Template?

Brightcove V-Commerce Template can provide an interactive video viewing experience by easily inserting clickable cue points into the video player in connection with the Video Cloud.

In the past, you had to develop interactive features for the video separately by using a Player API, but now you can produce a clickable video simply by inserting images, landing page URL, basic description of the product, or price using four default styles we provide through V-Commerce Template without going through a development process.

V-Commerce Template can be used for commerce and also interactive video marketing. You can induce call-to-action such as connecting the video to SNS sites such as Facebook and Twitter, provide additional information related with the video, or download questionnaires and white papers through the Cue Point. Also, the Cue Point action written in JavaScript will work in the mobile environment as it would in the web environment, thus expanding your mobile commerce and marketing effect.

This document was prepared to explain how to use V-Commerce Template in combination with Video Cloud, the basic directions about using the template, and how to configure the template environment.

Thank You.

Brightcove Korea Team

Table of Contents

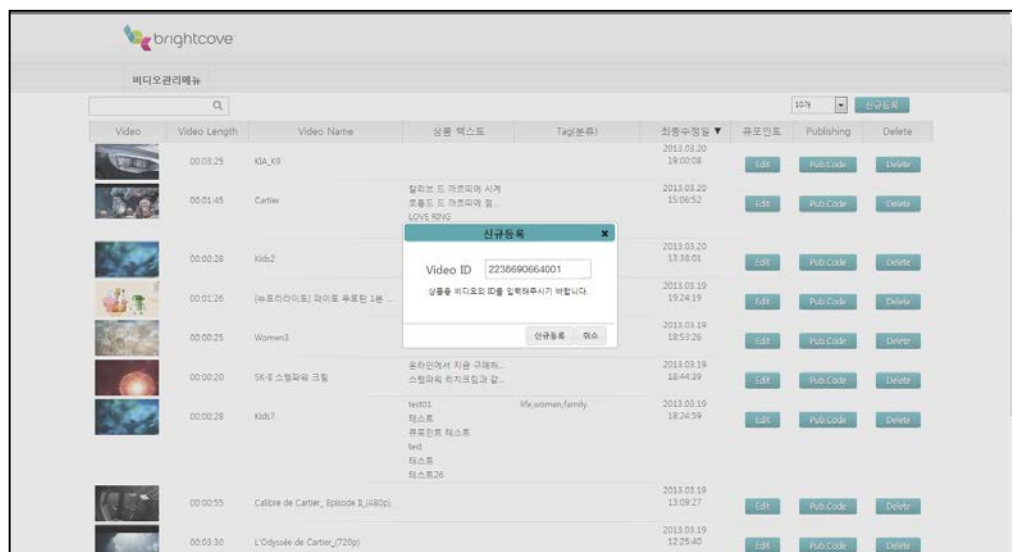
1. Installing Template & Using in Combination with Video Cloud	4
1.1 Template Structure: Admin Module + Player Module	4
1.2 Basic Installation Process	5
1.2.1 Installing Web Server: Install Apache webserver and PHP5.	5
1.2.2 Installing Compressed Files	5
1.2.3 Configuring Basic Environment of Admin Module	5
1.2.4 Configuration of the Basic Environment of Player Module	7
1.3 Changing the Size of Video Player	8
1.4 Admin Login Information Setting	8
2. How to Use Template.....	9
2.1 Admin (Administrator's Page)	9
2.1.1 Description of Admin (Administrator's Page) Menus	9
2.1.2 Uploading a New Video	10

2.1.3 Overview of Cue Point Type A, B, C, and D	12
2.1.4 Setting Cue Point	14
2.2 Player (Video Player Page)	17
2.2.1 Description of Player Menu.....	17
2.2.2 Sharing with SNS.....	17
2.3 Publishing	18
2.3.1 Embedding Video with Publishing Code	18
2.3.2 Video List Portal Publishing	19
2.3.3 Publishing Video Portal through Facebook Fan Page App	20
3. Configuration of Environment (config)	23
3.1. config Environment Configuration	23
3.1.1 Admin Environment Configuration	23
3.1.2 config Player Environment Configuration	25
3.1.3 Setting Google Analytics.....	29
3.2 CSS Environment Configuration	32
3.2.1 CSS Admin Environment Configuration.....	32
3.2.2 CSS Player Environment Configuration	32
4. Role of File and Function List.....	34
4.1 Admin	34
4.2 Player.....	35
5. Google Closure Compiler.....	39
6. FAQ.....	41
Q.1. How can I load the thumbnail image URL used for Cue Point?.....	41
Q.2. Why does it take 3-5 minutes to apply the settings I made to the template?.....	41
Q.3. I want to use other types of cue points other than those set in the template by default, how can I add them?.....	41
Q.4. Can I use the clickable videos with Video Cloud Express version?.....	41
Q.5. Can I play the clickable video in my mobile device?	41

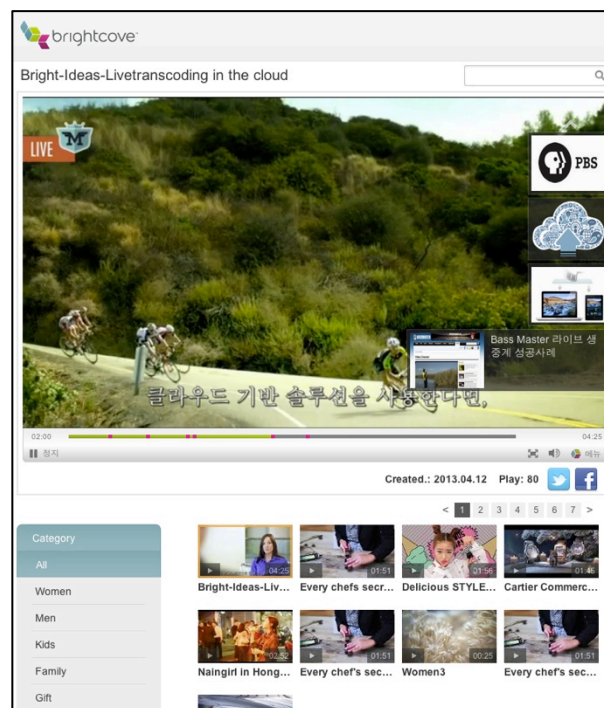
1. Installing Template & Using in Combination with Video Cloud

1.1 Template Structure: Admin Module + Player Module

1.1.1 Admin Module: Administrator's page for development and management



1.1.2 Player Module: Video player publishing and video list page composition



1.2 Basic Installation Process

1.2.1 Installing Web Server: Install Apache webserver and PHP5.

1.2.2 Installing Compressed Files:

- ① Designate the root folder for web services for player and admin.
- ② Decompress admin.zip file of the template into the admin folder.(Ex: /admin)
- ③ Decompress player.zip file of the template into the player folder. (Ex: /player)

1.2.2 Checking Media APIToken: Check the token in Account Settings ->API management of the Video Cloud Studio. Check Read or Write Type in API management screen as seen below to copy and use the key.

Account Settings

API Management

Token	Type (?)	Options (?)	Manage
[Redacted]	read	URL Access	[Icons]
[Redacted]	read		[Icons]
[Redacted]	write		[Icons]

You are using 2 of 6 available Read Tokens and 1 of 1 available Write Tokens

Add a:

Help Tips:

- Tokens are used to authenticate your request to the Media APIs. You cannot use the Media APIs without valid tokens for your account.
- Read tokens provide read-only access to your Brightcove Media Library.
- Read tokens with URL access enable you to access the media files themselves and not just the metadata.
- Write tokens give you access to create, modify, and delete videos and playlists in your Brightcove Media Library. [Learn More](#)

1.2.3 Configuring Basic Environment of Admin Module:

Configuring Media API token in PHP: Configure Video Cloud Token in ./api/config.php file. The following configuration is used to read or modify the video information of Video Cloud in admin.

```
<?PHP
```

```
$apiUrl = "http://api.brightcove.com/services/library";  
  
$writeApiUrl = "http://api.brightcove.com/services/post";  
  
$readToken = "Insert Read token";  
  
$writeToken= "Insert write token";
```

```
?>
```

Configuration of ./js/config.js environment

Set the default language. : en-English, ko-Korean

Register the path of the player folder to use when publishing the video player.

Insert Read Token to read the video metadata.

```
varLanguageCode = 'en'; // Korean(ko), English(en)

varPreviewBaseUrl = "http://www.mydomain.co.kr/ec/player/";

varBrightCoveAPI= {

    CMTV_Tag: "cm_tv",

    ApiBaseUrl: "http://api.brightcove.com/services/library",

    Token: "Insert Read token";

    PageSize: 10, // Choose the page size - 10,15,20,25,30

    SearchFields:
    "id,name,shortDescription,creationDate,playsTotal,videoStillURL,length,thumbnailURL,renditions,tag
    s,customFields,cuePoints,referenceId,lastModifiedDate",

    MediaDelivery:"http"

};
```

Configuration of ./js/jLanguage.js for language setting: Modify each message to the language to be used.
(Currently Korean and English are available.)

```
_O.Message = {

    'Won_ko' : "원",

    'Won_en' : "$",

    'Video_ko' : "비디오",

    'Video_en' : "Video",

    ....
```

1.2.4 Configuration of the Basic Environment of Player Module:

Configuration of ./js/config.js Environment

Player: Configure the ID, key, width, and height of the single/chromeless player to be used by default. **PageSize** is the total number of videos to be exposed on the list in the video list page. Since the number of total pages changes according to the size of the player, set this to an appropriate value. (Maximum 100)

LanguageCode: Choose the language to use in the video list page. Ko/en

Token: Configure Media API Token to use when reading the information from the Video Cloud.

gald: Enter the account ID of Google Analytics that you wish to use.

```
Player: {  
    .....  
    /// SinglePlayer ID &SinglePlayer KEY ///  
    SinglePlayerID:"Single Player ID",  
    SinglePlayerKEY:"Player key",  
    /// ChromelessPlayer ID &ChromelessPlayer KEY ///  
    ChromelessPlayerID:"Chromless Player ID",  
    ChromelessPlayerKey:"Chromeless Player key",  
    ....    /// Player CSS Style    ////  
    width:600,//Player's Width  
    height:400,//Player's Height  
    ....    end:0  
    },  
    LanguageCode: 'ko',// Korean (ko), English (en)  
    BrightCoveAPI: {  
        Token: "Insert Read Token",//Token value should be same with token.php  
        PageSize: 9, // List Display Item Number  
        ....    },  
    Analytics: {  
        gald: "UA-XXXXXXXX-1"  
    }  
}
```

Configuratioin of Token in Token.phpfile: Configure the token in the same manner as in config.js.

```
<?php

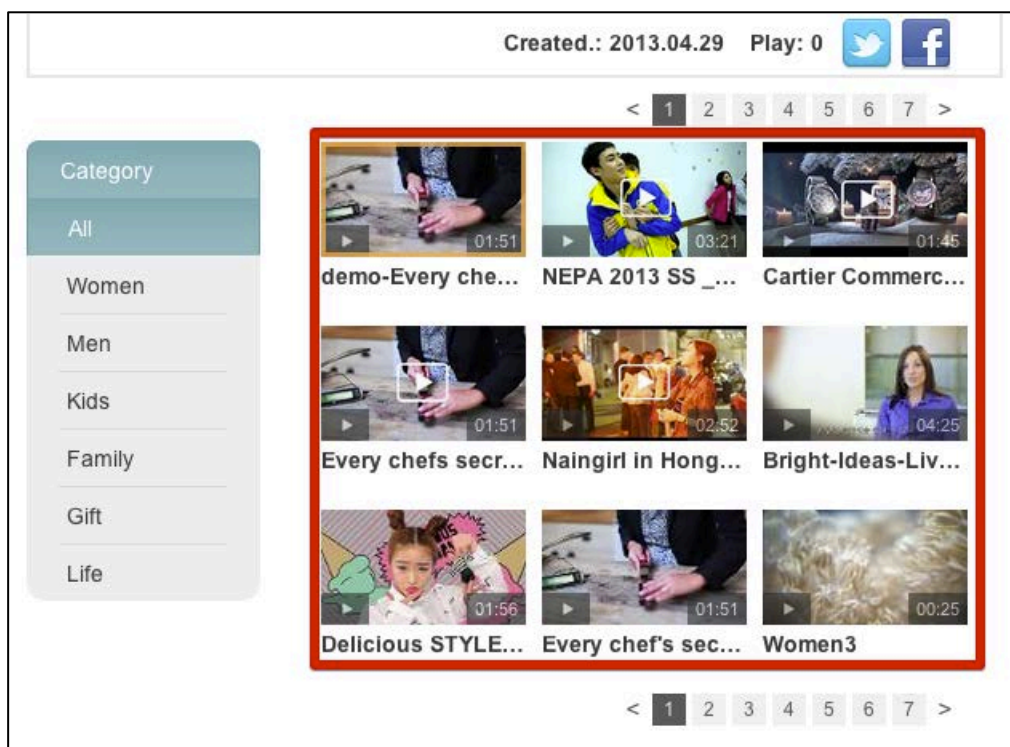
// Token Value should be equal to the token value under config.js

$token= "Set Read Token";
```

1.3 Changing the Size of Video Player

Change the width and height of the player in config.js of the player. Since the total number of pages in the video list page changes due to the size of the player, choose an appropriate size required for the service.

PageSize in Config.js is the total number of video lists. The number of videos in one line changes automatically according to the image size and the player width above, and the total number of videos in one page is the PageSize. In the following example, there are three lines (default) and the PageSize is 9.



1.4 Admin Login Information Setting

According to the characteristics of the template, the login ID and PW is set for the admin. If you wish to use login ID and PW as the basic setting before you manage security password with a separate system, you can modify ID and PW in ./index.php.

2. How to Use Template

2.1 Admin (Administrator's Page)

2.1.1 Description of Admin (Administrator's Page) Menu

Video	Video Length	Video Name	Description	Tags	Last Modified	Cue Point	Publishing code	Delete
	00:00:25	Women9		women	2012.12.07 11:25:06	Edit	Pub. Code	Delete
	00:00:25	Women2		women	2012.12.07 11:25:06	Edit	Pub. Code	Delete
	00:00:30	Men7		men	2012.12.07 11:24:40	Edit	Pub. Code	Delete
	00:00:30	Men8		men	2012.12.07 11:24:40	Edit	Pub. Code	Delete
	00:00:30	Men3		men	2012.12.07 11:24:40	Edit	Pub. Code	Delete
	00:00:30	Men10		men	2012.12.07 11:24:40	Edit	Pub. Code	Delete
	00:00:30	Men5		men	2012.12.07 11:24:40	Edit	Pub. Code	Delete
	00:00:30	Men9		men	2012.12.07 11:24:40	Edit	Pub. Code	Delete
	00:00:28	Kids8		kids	2012.12.07 11:24:05	Edit	Pub. Code	Delete
	00:00:28	Kids10		kids	2012.12.07 11:24:05	Edit	Pub. Code	Delete

- ① **Search:** You can search the same string in the tag name and video name.
- ② **Shows Video Thumbnail Image, Video Length, Video Name, Description** (Cue Point Name and List), **Tag, Last Modified Date, etc.**
- ③ **Sets the number of videos to be displayed on the same page.**
- ④ **Add New Video:** Opens a popup window for adding a new video.

Add new video ×

Video ID

Please insert Video ID

[Add new video](#) [Cancel](#)

- ⑤ **Edit:** Opens a popup window for editing the player type, cue point, and tag of the video.

Cue Point

Video Title: Bright-Ideas-Livetranscoding in the cloud

Video ID: 229451635001

Ref ID(optional):

Player Type: Single

Cue Point Type: CueType A : Thumbnail in right side

Tags in use:

Tag Library: women men kids family gift life

Buttons: Play video, Update, Cue Point, Refresh

- ⑥ **Pub.code:** You can copy the publishing code of the video.

Pub.Code

```
<div class="BrightCoveCT"></div><script>
(function(J) {
    window
    ["g_br_api_url"] = J.src; var s =
    J.doc.createElement("script"); s.type
    = "text/javascript"; s.async = true;s.src =
    J.src; var s0 =
    J.doc.getElementsByTagName("script")[0];
    s0.parentNode.insertBefore(s, s0;)}
    ({src:"http://www.jnjl.co.kr/prj/Brightcove/pla
    yer_en/js/api.js?
    id=2233975588001",doc:document});</scrip
    t>
```

Buttons: Select, Cancel

Text: Press Ctrl-C copy to clipboard

- ⑦ **Delete:** Delete a video.

- ⑧ **Page Navigation**

2.1.2 Uploading a New Video

To create a video cue point using the commerce template, you need to add the ID of the video uploaded to the Brightcove Video Cloud Studio as 'New Video'. Find the ID of the video to insert the clickable feature from the Video Cloud Studio and enter the ID to the Add New Video window of the VCommerce Template.

[New Video ID Input Window]

Add new video

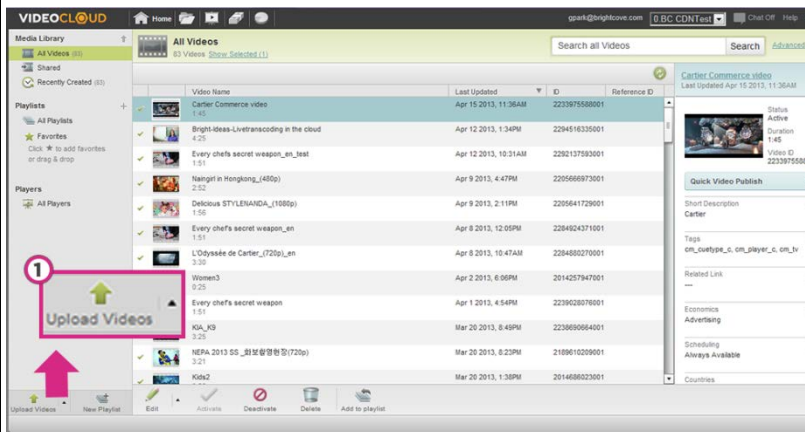
Video ID: [Input Field]

Please insert Video ID

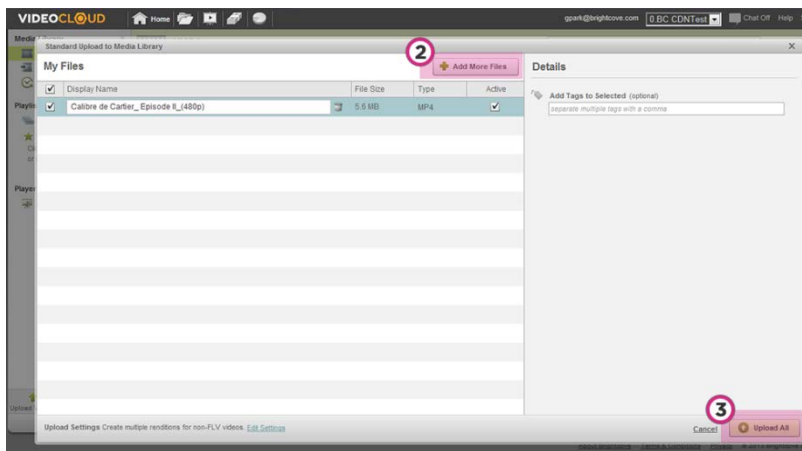
Buttons: Add new video, Cancel

*Video ID is made of a 13-digit number. Refer to the following for finding Video IDs in the Video Cloud Studio.

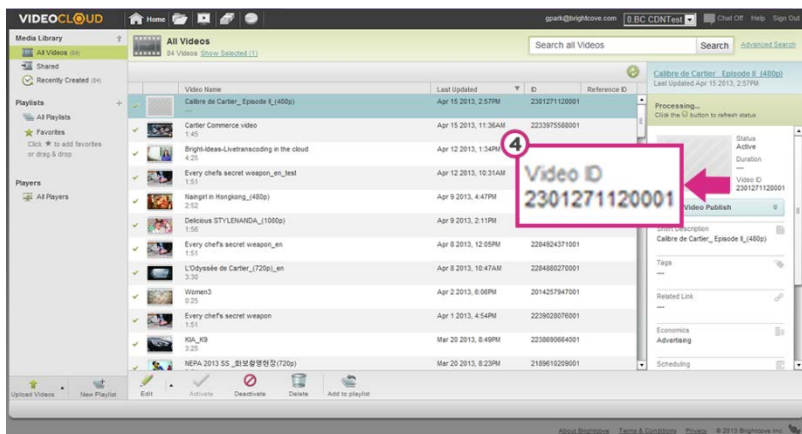
[How to Get Video ID from Brightcove Video Cloud]



1. Log on to Brightcove Studio, select 'Upload Video' from 'Media' menu.



2. Choose the video to upload with 'Add New Video'



3. After selecting the video, press the 'Upload All' button to start uploading.
4. You can check 'Video ID' in the detailed information area on the right when you click the uploaded video.

2.1.3 Overview of Cue Point Type A, B, C, and D

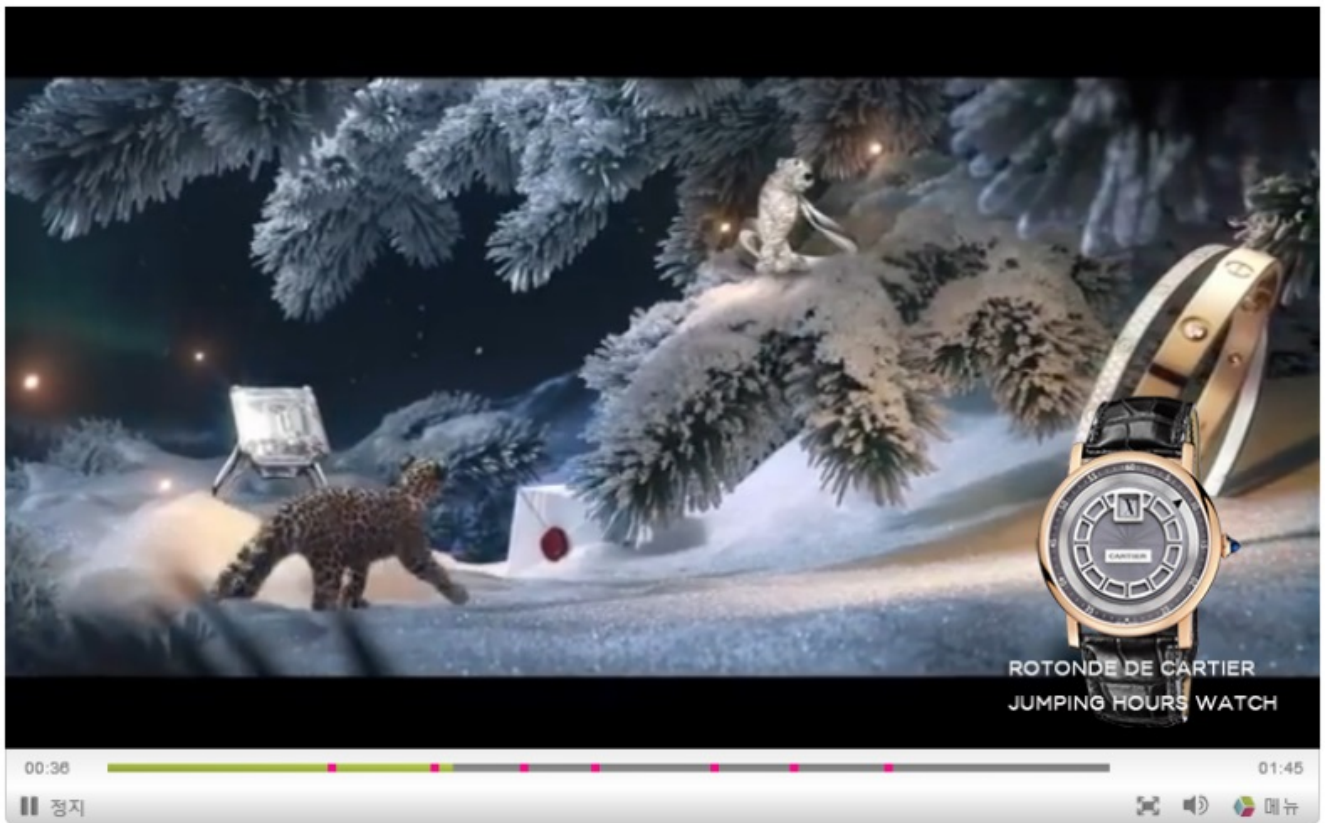
Cue Type A: Thumbnail on the right



Cue Type B: Banner on the bottom



Cue Type C: Banner on the bottom left and right

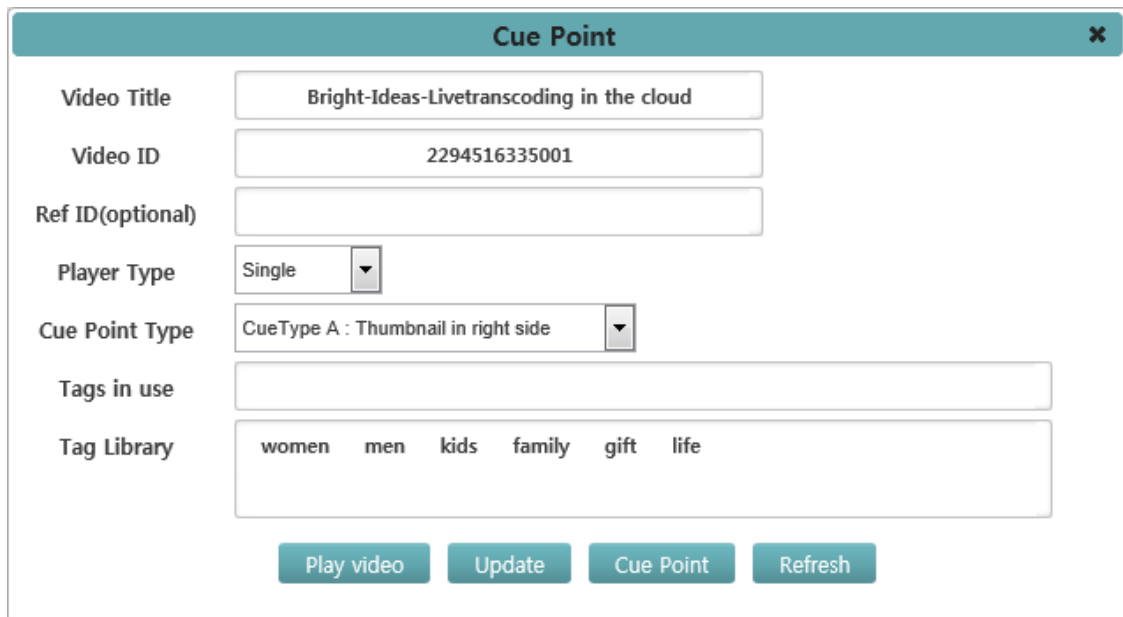


Cue Type D: Banner on the bottom (Close)



2.1.4 Setting Cue Point

2.1.4-a. Cue Point Configuration Menu



- ① **Video Title:** Title of the video (Name of the video uploaded to Video Cloud)
*You can modify the video title in Video Cloud Studio.
- ② **Video ID:** ID of the video registered in the Video Cloud
- ③ **Ref ID(Optional):** Reference ID of the corresponding video
*What is a Reference ID?
: Reference ID is used to work with other services such as the publisher's contents management system apart from the Video ID which is automatically generated by Video Cloud. Click [here](#) for details.
- ④ **Player Type:** Player type of the corresponding video. Choose between Single Player and Chromeless Player.



- ⑤ **Cue Point Type:** Select cue point type of the corresponding video
Cue Type A: Thumbnail on the right
Cue Type B: Banner on the bottom
Cue Type C: Banner on the bottom left and right
Cue Type D: Banner on the bottom (Close)
- ⑥ **Tags in Use:** Tags currently registered in the corresponding video
- ⑦ **Tag Library:** Tags that can be registered in the video


- ⑧ **Play Video:** Preview of the video with the cue point applied to the video
- ⑨ **Update:** Apply the items (player type, cue point type, tag setting) set above to the video
- ⑩ **Cue Point:** Opens cue point management (add/delete/modify) window
- ⑪ **Refresh:** Refreshes items currently displayed
 - According to the characteristics of the cloud server, the update will be displayed normally after 5 to 15 minutes.

2.1.4-b. Cue Point Modification Order

1. Select the type of player between Single or Chromeless.
2. Select the cue point type to use from the drop down menu.
 - Cue Type A: Thumbnail on the right
 - Cue Type B: Banner on the bottom
 - Cue Type C: Banner on the bottom left and right
 - Cue Type D: Banner on the bottom (Close)
3. Click to set the tag to apply to the video. You can click the tag to apply to the video in the Tag Library to set the tag to the video.
 - *You can see the videos categorized by their tags in the Player list page.
4. Click cue point button to edit the cue point.

[Cue Point Management Window]

Cue Point Settings ✕



Timecode(hh:mm:ss)

Align, Bottom

Left

Alert: Bottom value cannot exceed 796 (Height length of image)

Item ID

Item Price(\$)

Description

Thumbnail URL

Overview Thumbnail

The thumbnail size of Type A is 100x75,
Type B is 450x75

Link URL

Opens in a new window ☐

Save

Cancel

- ① **Time Code:** Determine when the cue point will appear. Use the play/stop button of the player on the top.
- ② **Item ID:** You can add product ID.
- ③ **Item Price (\$):** Enter the price of the product to be displayed at the cue point.
*You can use only numbers to enter the price. The comma (,) will be automatically added for the display.
- ④ **Description:** Enter the description of the product to be displayed at the cue point.
- ⑤ **Thumbnail URL:** Enter the URL of the thumbnail to use at the cue point.
*Click View Thumbnail to view the thumbnail.
- ⑥ **Link URL:** Enter the landing page URL to connect to when the user clicks the cue point.
- ⑦ **Open in New Window:** Decide whether the landing page will be opened in a new window or within the player when the user clicks the Cue Point.
- ⑧ **Align, Bottom:** This appears for Cue Point Type C only. You can determine the direction the Cue Point appears (left/right) and the height of the Cue Point (set the distance from the bottom).

*Cue Point Types C and D do not display price and description but only the image.

* The thumbnail size of Cue Point Type A is 100*75, Type B is 450*75, and Type C is free.

* PNG image files with transparent background are recommended for Cue Point Type C.

5. Save Cue Point Setting

Add all cue points to set and press the 'Save' button to save cue point setting.

큐포인트 설정 ✕



큐포인트 추가

순서	큐포인트 썸네일	시간	상품 텍스트	상품 가격	변경	재생
1		00:00:11	3 CONCEPT EYES LIP COLOR-#504	17,900 원	수정 삭제	미리보기
2		00:00:18	3 CONCEPT EYES NAIL LACQUER #OR01	4,000 원	수정 삭제	미리보기
3		00:00:27	3 CONCEPT EYES ONE COLOR SHADOW(...	11,000 원	수정 삭제	미리보기
4		00:00:47	3 CONCEPT EYES GEL EYE LINER #GLITTE...	12,000 원	수정 삭제	미리보기

저장

큐포인트 설정 ✕

큐포인트 추가

순서	큐포인트 썸네일	시간	상품 텍스트	상품 가격	변경	재생
1		00:00:11	3 CONCEPT EYES LIP COLOR-#504	17,900 원	수정 삭제	미리보기
2		00:00:18	3 CONCEPT EYES NAIL LACQUER #OR01	4,000 원	수정 삭제	미리보기
3		00:00:27		11,000 원	수정 삭제	미리보기
4		00:00:47		12,000 원	수정 삭제	미리보기

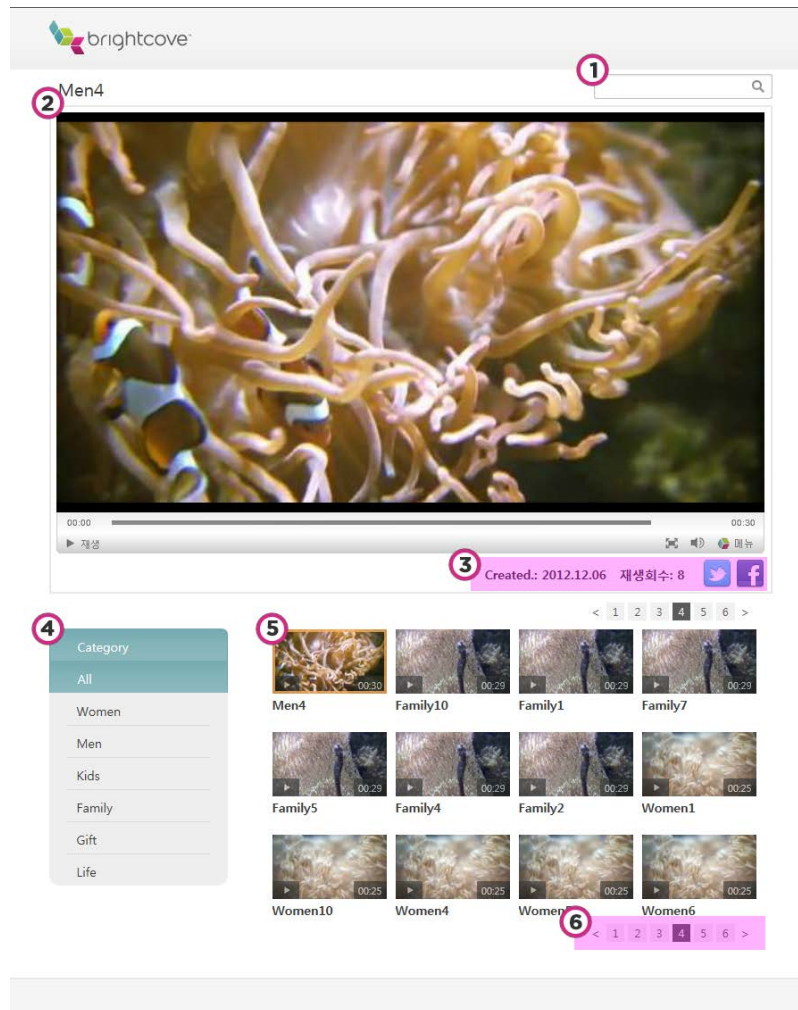
Save ✕

저장하시겠습니까?

OK
Cancel

2.2 Player (Video Player Page)

2.2.1 Description of Player Menu



- ① **Search:** Searches for the search string in the tag name and video title.
- ② **Video Play Area:** This is the area that plays the selected video.
- ③ **Video Upload Date, Number of Views, SNS Share Button Area.**
- ④ **Categorizes videos in the list by their tags.**
*Category can be added or modified by revising config.js.
- ⑤ **Videos Played:** Highlights the video currently activated.
- ⑥ **You can divide the video list into pages.**

2.2.2 Sharing with SNS

Created.: 2013.03.06 재생회수: 43

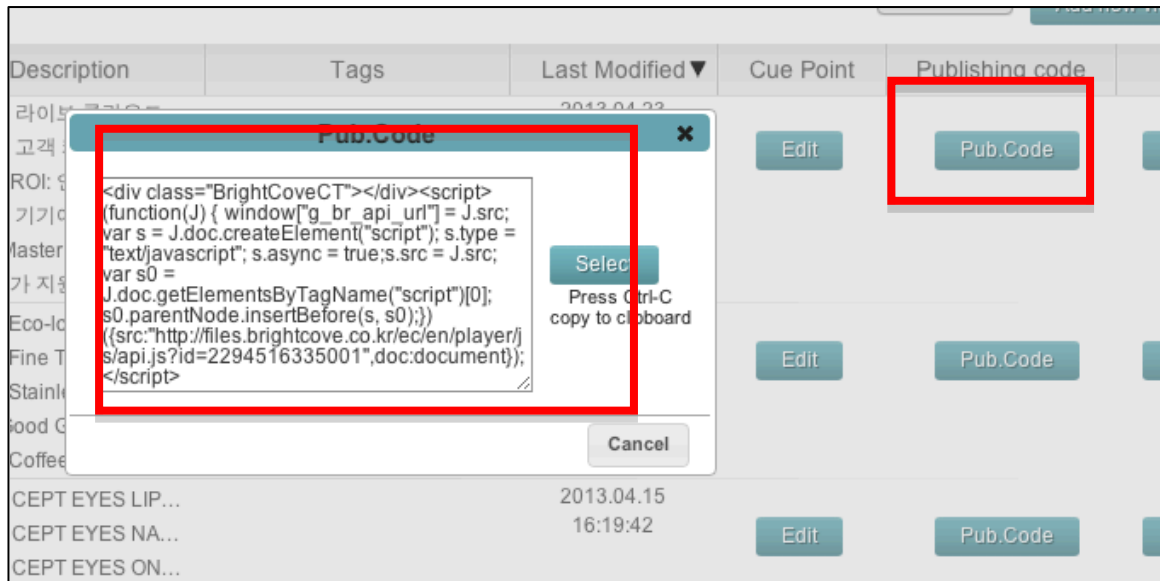


You can share the clickable video through SNS using the Twitter and Facebook buttons located on the bottom of the player. SNS Share Button can be turned on or off in config to display them on the bottom of the player or to remove them.

2.3 Publishing

2.3.1 Embedding Video with Publishing Code

You can insert the video to an HTML web page, blog, or microsite with the publishing code of single clickable video player. Click Pub.Code button of the video to publish in admin to see the publishing code of the corresponding video.



Copy and paste the said publishing code to the web page to service to publish the video player within the page.

Using this method, you can publish your videos in individual websites as follows.

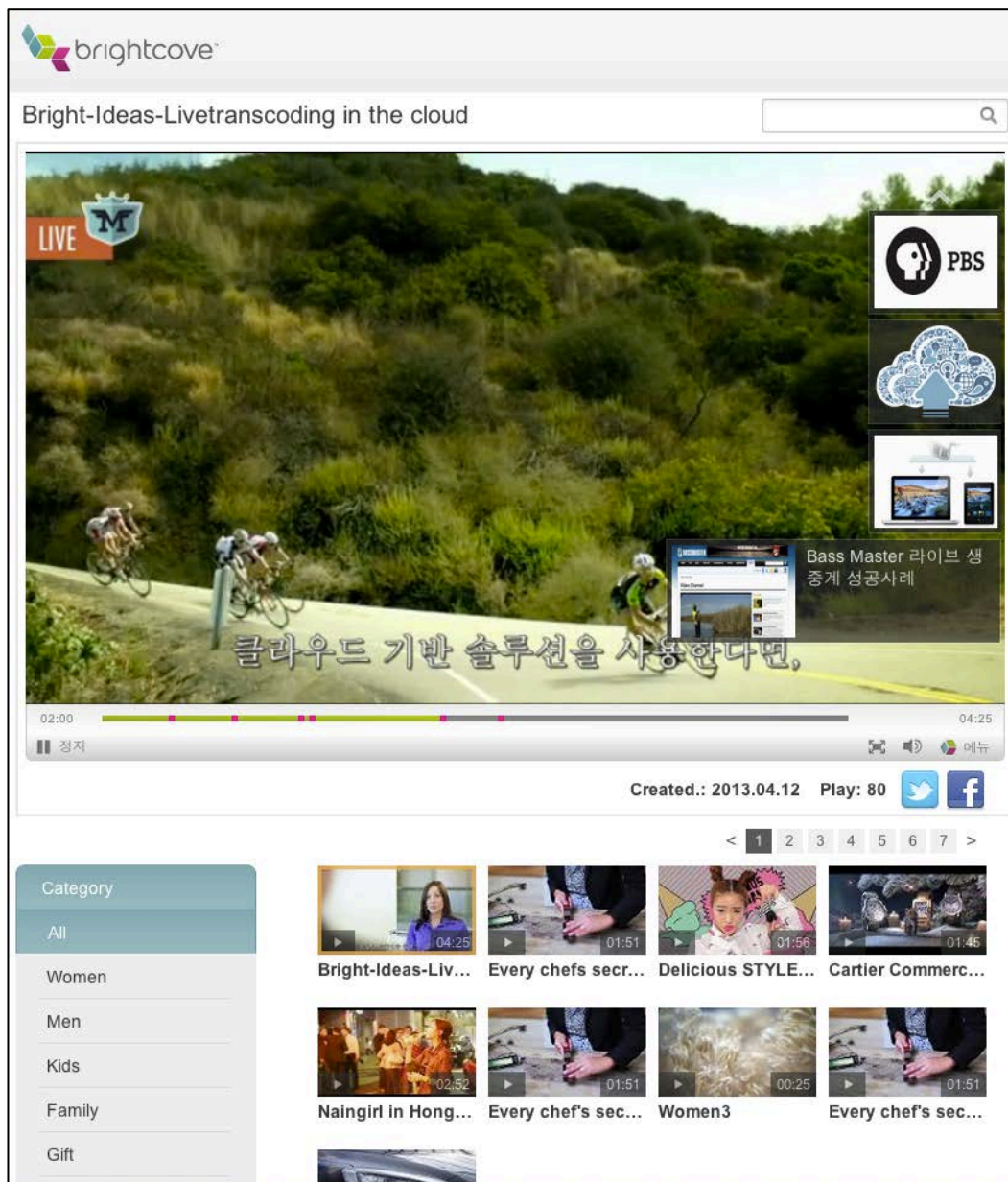


2.3.2 Video List Portal Publishing

When you install the player module and open the corresponding index.php, you can see the portal screen to service video player and video list.

Modify environment setting and CSS adjustment to change your videos to the format you want and add them to existing web services.

The default size of the video player is set to 800x600, and you can change the size to your preference.



2.3.3 Publishing Video Portal through Facebook Fan Page App

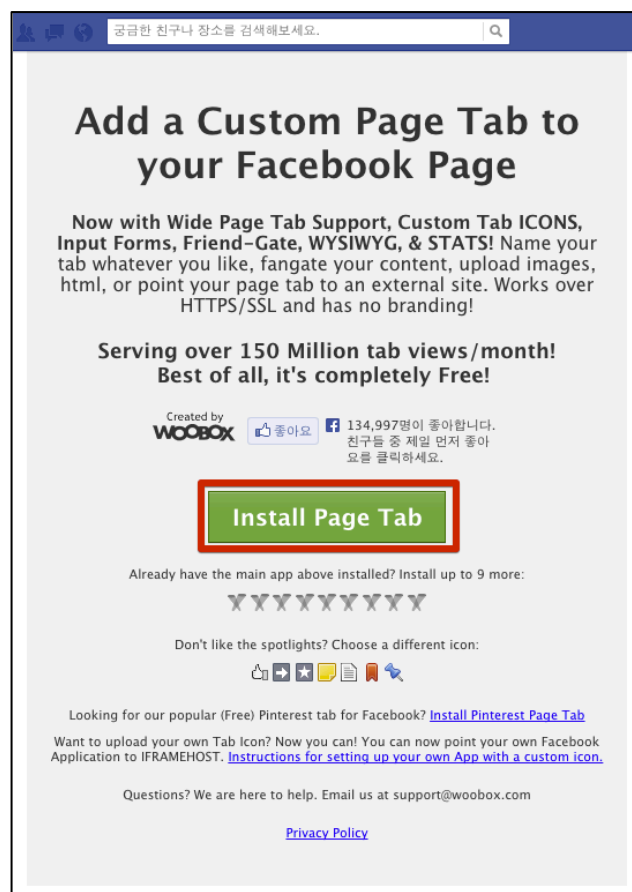
You can publish the video portal you installed through the player module to the Facebook FanPage using an app.

First, make a Facebook Page, and add an app for Facebook through iframe.

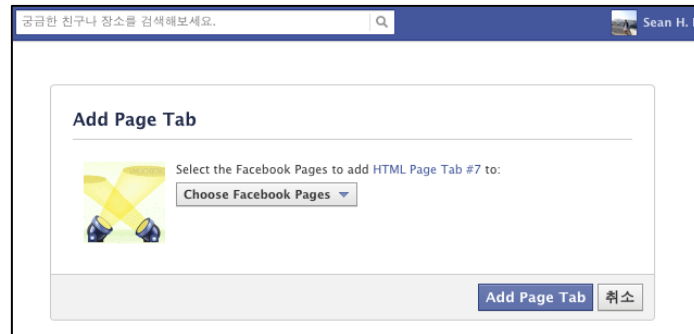
: Enter iframe in the search window of the Page and select the displayed iframe app to install.



This example will use the first Woobox for publishing. Select woobox from the top and select Install Page Tab from the app page in the following.






Click the Install Page Tab to open the screen for selecting a page as follows. Select a page to add the app to the page.



The app added will appear as follows.



When you open the app here, you can see the settings page for the app as follows. Select “URL” type beside Page Source and enter the address of the player module you prepared to the address window. Set the name of the tab and change image to complete the registration of the app.

Tab Settings   

Page Tab Share Settings

Brightcove.co.kr [Edit](#)
Visit Brightcove.co.kr
Shareable Tab URL: <http://statictab.com/7xkz97s> ☐ Show Share button at top of Tab

Page Source: ☒ URL ☐ Redirect ☐ Image ☐ HTML

Enter a URL to load in the iframe window Height px
☐ Hide Scrollbars

Fan-Gate (Non-Fan Page Source): ☒ Off ☐ URL ☐ Image ☐ HTML
Fan-gating is turned off. Fans and Non-fans will see your tab content.








Friend-Gate: ☒ Off ☐ Default ☐ HTML
Friend-gate is turned off. Visitor's friends do not need to be fans to access your content.

Form-Gate: ☒ Off ☐ HTML
Form-gate is turned off. Visitors do not complete an input form to access your content.

Tab Icon: Instructions for using your own tab icon.

Tab Name:

Tab Image: [Change Tab Image](#) (111px wide by 74px tall)

Install Another Tab:       

[Cancel](#) [Save Settings](#)


Click Save Settings and save to see the screen published.

회원님은 Brightcove.co.kr 이름으로 게시, 댓글 달기, 좋아요 클릭을 하고 있습니다. — Sean H. Park(으)로 변경

Brightcove.co.kr EC Template 좋아요



brightcove

Cartier Commerce video



00:35 01:45

정지

등록일: 2013.03.18 재생회수: 576  

< 1 2 3 4 5 6 7 >

Category

- All
- Women
- Men
- Kids
- Family

demo-Every che... 01:51

NEPA 2013 SS _... 03:21

Cartier Commerc... 01:45

Every chefs secr... 01:51

Naingirl in Hong... 02:52

Bright-Ideas-Liv... 04:25

3. Configuration of Environment (config)

3.1. config Environment Configuration

3.1.1 Admin Environment Configuration

Configure environment using config.js.

[config.js]

```

vargTagsName= {
    ct_tags: ["ct_Women", "ct_Men", "ct_Kids", "ct_Family", "ct_Gift", "ct_Life"]
};

vargCuePointType = {
    "*" : " CueType* : CueType* ",
    "A" : " CueTypeA : Thumbnail on the right ",
    "B" : " CueTypeB : Banner in the bottom ",
    "D" : " CueTypeD : Banner in the bottom(Close button) "
};

vargCuePointPos = {
    "L" : "Left",
    "R" : "Right"
};

vargPlayerType = {
    "S" : "Single",
    "C" : "Chromeless"
};

vargLanguageCode = 'en'; // Korean(ko), English(en)

vargPreviewBaseUrl = "http://www.jnjt.co.kr/prj/Brightcove/player/";
vargPubCode = {
    Player:
        '<div class="BrightCoveCT"></div>'+
        '<script>'+
            '(function(J) {                window["g_br_api_url"] = J.src; var s = J.doc.createElement("script"); s.type = "text/javascript";
s.async    =    true;s.src    =    J.src;    var    s0    =    J.doc.getElementsByTagName("script")[0];    s0.parentNode.insertBefore(s,    s0);})
({src:""+gPreviewBaseUrl+'js/api.js?id=#VIDEO_ID",doc:document});'+
            '</script>'
        };

vargBrightCoveAPI= {
    CMTV_Tag: "cm_tv",
    ApiBaseUrl: "http://api.brightcove.com/services/library",
    Token : "GjGg0W5wxXT5tRa0W1jtCsm1gDg411r9C4P5kbvUpN2OANQ6bdC4Cw..",
    PageSize: 10, // 10,15,20,25,30 중 택1.
    SearchFields:"id,name,shortDescription,creationDate,playsTotal,videoStillURL,length,thumbnailURL,renditions,tags,customFields,cuePoints,referenceld
        ,lastModifiedDate",
    MediaDelivery:"http"
};

```

1) gTagName

Designate category Tag. Use “ct_” as the prefix.

For your reference, the category name displayed on the player can be set in ctLanguage.js of the player.

In config.js of the player, it will be designated without the prefix, “ct_” in gConfigclass categories tags. Designate the exposure string of the corresponding ctLanguage.js designates.

config.js files

```
gConfig.Categories: {  
  
    tags: [ "all", "women", "men", "kids", "family", "gift", "life"]  
  
},
```

In Korea, ctCategory.koclass is used, and it is saved as follows.

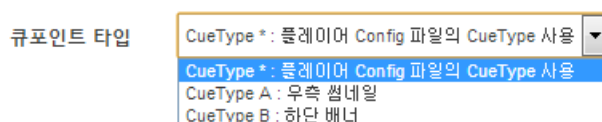
ctLanguage.koclass in **ctLanguage.js** file is as follows.

```
"CategoryTitle" : "Category",  
  
"all"             : "All",  
  
"women"          : "Women",  
  
"men"            : "Men",  
  
"kids"           : "Kids",  
  
"family"         : "Family",  
  
"gift"           : "Gift",  
  
"life"           : "Life",
```

In other words, if the category name is men, the category name exposed on the screen is “Men” on the right.

2) gCuePointType

Designate the type of Cue Point UI. This is used to designate the exposure type of the cue point on the video when creating and modifying the cue point. By default, it will be exposed in the type specified in the config file of the Player.



3) gCuePointPos

This is applied when the Cue Point type is C. This decides which direction the Cue Point image will appear on the player.

4) gLanguageCode

This is the variable for selecting the language (Korean, English) of Admin.ex) ko:Korean, en: English

5) gPreviewBaseUrl

This designates the http URL of the player’s location. This is used to load the Preview Player in Admin screen. This is the URL of the player used to insert the player into a web page.

6) gPubCode

This is the API code used to insert the player into a page. It calls js/api.js at the URL of the player designated by gPreviewBaseUrl. The player will be inserted between <div class=“BrightCoveCT”></div>. In other words, BrightCoveCTclass name must not be changed.

7) gBrightCoveAPI

Designate the variables used to call BrightCove API.

Variable name	Description
CMTV_Tag	Uses cm_tvtage, must not be modified.
ApiBaseUrl	BrightCove API URL.
Token	Read Token used to call BrightCove API. Must be modified to the user's token.
PageSize	Designates the initial page size of the video list. This can be set to 10, 15, 20, 25, and 30.
SearchFields	Designates the field parameter of the Search API of BrightCove
MediaDelivery	Designates BrightCove Media Delivery protocol. httporhttp_ios can be used.

3.1.2 configPlayer Environment Configuration

Configure environment in config.js.

[config.js]

```
vargConfig = {  
  Categories: {  
    tags: [ "all", "women", "men", "kids", "family", "gift", "life"  
  ],  
  SetCss: {  
    playerBorderColor:"#E8E8E8", // Player Wrapper Box Border Color  
    selectItemFocusColor:"#FAAC58", // List Select Item Focus Color  
    listItemImgWidth:128, // List's Items Image Width  
    listItemImgHeight:72, // List's Items Image Height  
    listItemNameHeight:28, // List's Items Title Box Height  
    listItemNameFontSize:14, //List's Items Title Font Size  
    listItemNameFontLineHeight:14, //List's Items Title Font Line-Height  
    listItemNameFontColor:"#3B3B3B", //List's Items Title Font-Color  
    listPageBoxOffBGColor:"#F1F1F1", // List Page Normal Background-color  
    listPageBoxOffFontColor:"#5B5B5B", //List Page Normal Font-color  
    listPageBoxOnBGColor:"#5B5B5B", //List Page Focus Background-color  
    listPageBoxOnFontColor:"#F1F1F1", //List Page Focus Font-color  
    logoBGColorImg:"url(/images/header-top.gif) repeat-x top left", //Logo Background Image Url  
    logoImgUrl:"url(/images/logo-corporate-new.png) no-repeat top left", //Logo Image Url  
    footerBGColorImg:"url(/images/footer.png) repeat top left", //Footer Image Url  
    //end//  
    end:0  
  },  
  Player: {  
    showInfoSNS: true, // show Player Info (Created date and Number of Play, and twitter and facebook  
    showPubCodeInfoSNS: false, //PubCode Player InfoSNS off ***** P.S ---- Don't Modify *****  
    showCuePoint: true,  
    showPlayerWrapper: true, // PubCode Player Wrapper Border on/off  
    CuePointType: "CueTypeA",  
  
    /// SinglePlayer ID &SinglePlayer KEY ///  
    SinglePlayerID:"2057499176001",  
    SinglePlayerKEY:"AQ~~,AAAA_jYwAak~,f6iN_Qpszc3ypDaxWlg_4aUIMKWfwWeY",  
    /// ChromelessPlayer ID &ChromelessPlayer KEY ///  
    ChromelessPlayerID:"2079935948001",  
    ChromelessPlayerKey:"AQ~~,AAAA_jYwAak~,f6iN_Qpszc24AqdlmhbfBquNHNBSPFRO",  
  
    //// Player CSS Style ////  
    width:700, //Player's Width  
    height:450, //Player's Height  
  
    /// Cue Point Text ///  
    CuePointFontSize:14, //CueTypeA&CueTypeB Cue Point Font-Size  
    CuePointFontLineHeight:18, //CueTypeA&CueTypeB Cue Point Font Line-Height  
  
    ///CueTypeA///  
    aTypeCuePointBoxTop:30, //CueTypeA'sCuePoint Display Box Wrapper Top Position
```

```

        ///CueType B///

        ///CueType C///

        ///CueType D///

        //end//
        end:0
    },
    LanguageCode: 'ko', // ko and en
    CuePoint: {
        CueTypeA: {
            animationSpeed: 400, // animation moving time (ms):
            showTime: 2000 // Show time
        },
        CueTypeB: {
            animationSpeed: 1200, // animation moving time (ms):
            showTime: 10000 // Show time
        },
        CueTypeC: {
            animationSpeed: 1200, // animation moving time (ms):
            showTime: 10000 // Show time
        },
        CueTypeD: {
            animationSpeed: 1200, // animation moving time (ms):
            showTime: 10000 // Show time
        }
    },
    ADPopup: {
        width: 900,
        height: 600
    },
    BrightCoveAPI: {
        CMTV_Tag: "cm_tv",
        ApiBaseUrl: "http://api.brightcove.com/services/library",
        Token : "GjGg0W5wxXT5tRa0W1jtCsm1gDg411r9C4P5kbvUpN2OANQ6bdC4Cw.",
        PageSize: 12,
        SearchFields:
    "id,name,shortDescription,creationDate,playsTotal,videoStillURL,length,thumbnailURL,renditions,tags,customFields,cuePoints,referenceId",
        MediaDelivery:"http"
    },
    Analytics: {
        gald: "UA-35744930-1",
        appName: "jBrightCovePlayer"
    }
};

```

```

////// GA log //////////////////////////////////////
var _gaq = _gaq || [];
_gaq.push(['_setAccount', gConfig.Analytics.gald]);
// _gaq.push(['_trackPageview']);
(function() {
    varga = document.createElement('script'); ga.type = 'text/javascript'; ga.async = true;
    ga.src = ('https:' == document.location.protocol ? 'https://ssl' : 'http://www') + '.google-analytics.com/ga.js';
    var s = document.getElementsByTagName('script')[0]; s.parentNode.insertBefore(ga, s);
})();
////////////////////////////////////

```

```

gConfig.Categories.func = {
    Parent: gConfig.Categories,

    getTag: function (idx) {
        return (this.Parent.tags[idx]?this.Parent.tags[idx]: "");
    }
};

```

```


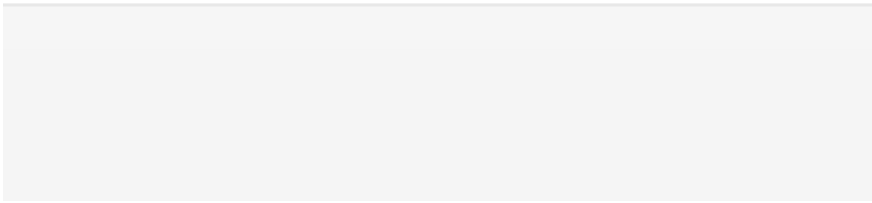

},
getIdx: function (name) {
    if(typeof name == 'undefined') return -1;
    var tags = this.Parent.tags;
    rv = 0; // default is first
    for(vari=0; i<tags.length; i++) {
        if(name.toLowerCase() == tags[i].toLowerCase()) {
            rv = i;
            break;
        }
    }
    returnrv;
}
};

```

1) gConfigClass Variable

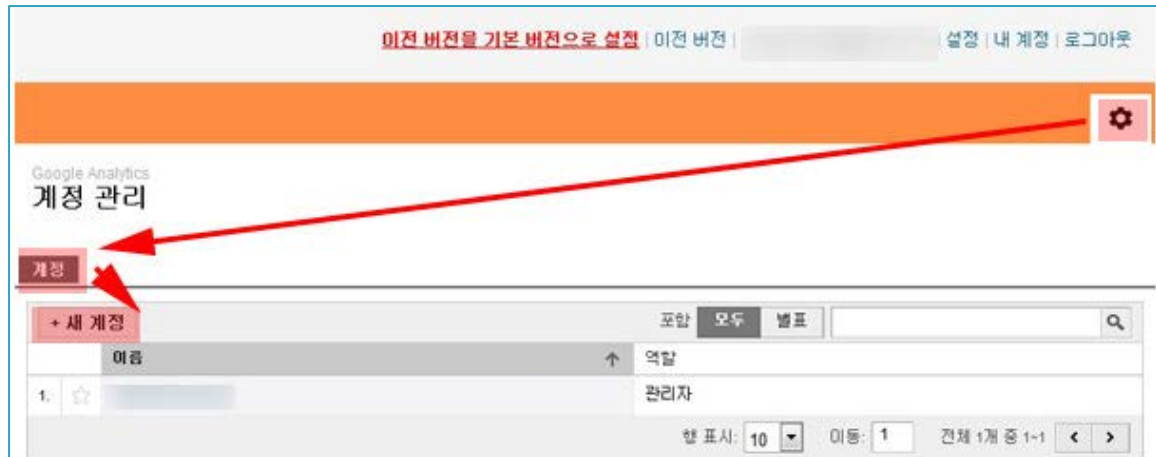
Set global variables.

Variable Name	Description
Categories	Designate category tag. all is not the tag used by Brightcove, however, is used to search all videos without a tag.
SetCss	<p>A set of variables called by js, page UI setting.</p> <p>playerBorderColor is the color of the border of the WrapperBox that wraps the player.</p>  <p>selectItemFocusColor is the color of the focus that appears when the list item is selected.</p> <p>listItemImageWidth is the width of the list item image.</p> <p>listItemImageHeight is the height of the list item image.</p>  <p>listItemNameHeight is the height of the name box of the list item.</p> <p>listItemNameFontSize is the font size of the name of the list item.</p> <p>listItemNameFontLineHeight is the line-height of the font of the name of the list item, and it is recommended that it is set to the same value as listItemNameFontSize.</p> <p>listItemNameFontColor is the font color of the name of the list item.</p>  <p>listPageBoxOffBGColor is the normal background color of the list page.</p> <p>listPageBoxOffFontColor is the normal font color of the list page.</p>  <p>listPageBoxOnBGColor is the background color of the list page when focused.</p> <p>listPageBoxOnFontColor is the font color of the list page when focused.</p>  <p>logoBGColorImg is the URL of the layer image on the top including the logo.</p> <p>logoImgUrl is the URL of the logo.</p>

	 <p>footerBGColorImg is the URL for the background image of the footer at the bottom of the list.</p> 
Player	<p>This is the variable for player UI setting.</p> <p>showInfoSNS determines the exposure of the UI on the bottom of the player. If True, SNS and play information area indicated in red in the following will be exposed and if False, the area will not be exposed.</p>  <p>showCuePoint determines whether the Cue Point UI shall be exposed on the player.</p> <p>showPubCodeInfoSNS can be turned on or off in the config.js of SNS in the screen which embeds the publishing code and gConfig.Player.showPubCodeInfoSNS. *****/This is not modifiable. Do not modify. *****/</p> <p>showPlayerWrapper determines whether the Player's Wrapper Border should be On/Off when publishing through the publishing code.</p> <p>CuePointType indicates that the cue point of the video is Cue Type* in Admin and is used to determine the Default Cue Point exposure type when the Player's config setting is available for use. Currently, "CueTypeA", "CueTypeB", "CueTypeC" and "CueTypeD" are available.</p> <p>SinglePlayerID is the player's ID. The user needs to modify this with his/her own player ID.</p> <p>SinglePlayerKEY is the player's key. The user needs to modify this with his/her own player key.</p> <p>ChromelessPlayerID is the same as the SinglePlayerID above</p> <p>ChromelessPlayerKey is the same as the SinglePlayerKEY above</p> <p>Width is the player's width</p> <p>Height is the player's Height</p> <p>CuePointFontSize is only applied to Cue PointA and Cue PointB, and it is the size of the Cue Point text.</p> <p>CuePointFontLineHeight is applied to Cue PointA and Cue PointB, and it is the value of the line-height of the font for the Cue Point text.</p> <p>aTypeCuePointBoxTop is only applied to Cue PointA, and it is the value of the top side of the BoxWrapper that wraps the Cue Points displayed.</p>
LanguageCode	Language setting variable. Currently 'ko' for Korean and 'en' for English can be used. This is used to choose the corresponding language in ctLanguage.js.
CuePoint	Designates the setting for Cue Point type. animationSpeed sets the sliding speed of the banner. showTime determines how long the banner should be exposed.
AdPopup	Width and height of the window overlaid when the user clicks Cue Point.
BrightCoveAPI	Designates settings for BrightcoveSearch API. Set Token value, etc.
Analytics	This is for Google Analytics, set gaid to Google GA ID.

3.1.3 Setting Google Analytics

1. You must have a Google account in order to use Google Analytics. Log on using the Google account.
2. Log on and select the settings button. Click Account + New Account, and register your blog and site.
3. Set it to your site and blog.



계정

새 계정 만들기
추적하려는 사이트의 URL을 입력하고 Google 웹로그 분석 보고서에 표시할 이름을 지정하십시오. 2개 이상의 웹사이트를 추적하려면 계정이 설정된 후 사이트를 추가하십시오. ?

기본 정보

계정 이름:

웹사이트 URL: (예: http://www.mywebsite.com)

시간대: (그리니치 표준시 +09:00) 서울

데이터 공유 설정 ?

공유 설정 ☐ 내 Google 웹로그 분석 데이터 공유 안 함

☒ Google 웹로그 분석 데이터를 공유...

☒ 다른 Google 제품과만 공유 선택사항
웹사이트의 Google 웹로그 분석 데이터를 다른 Google 서비스와 공유하여 애드워즈, 애드센스 및 기타 Google 제품의 광고 기능을 개선하고 사용자 만족도를 높이십시오. Google 서비스와 사용 (타사 제공) 데이터에 액세스할 수 있습니다. [예 보기](#)

☒ Google 및 기타 서비스와 익명으로 공유 선택사항
웹사이트 데이터를 익명으로 공유하여 벤치마킹 서비스를 이용하실 수 있습니다. Google의 벤치마킹 서비스는 웹사이트에 대한 개인 식별 정보를 모두 제거한 후 데이터 집계에 속한 수백 개의 익명 사이트 데이터와 조합하여 최근 업계 동향에 대한 통계를 집계합니다. [예 보기](#)

사용자 계약
아래에서 거주 중인 국가 또는 지역을 선택하십시오. 이용약관을 자세히 읽은 후 동의할 경우 확인란을 선택하십시오.

거주 국가 또는 지역:

GOOGLE ANALYTICS 서비스 약관

다음은 본 약관에 명시된 Google Analytics 서비스 ("본 서비스") 사용에 관한 Google Inc. ("Google")와 사용자 (이용자가 수권 직원 또는 대리인의 자격으로 대표하는 개인 또는 법적 주체) 간의 이용 조건입니다. 다음 내용을 숙독하여 주십시오. 이용자는 "동의함" 버튼을 클릭하거나, 등록 절차를 완료하거나, 본 서비스를 사용함으로써, 자신이 계정을 취득할 자격이 있고 본 서비스 이용 약관 전체 ("본 약관")에 구속되는 웹사이트 방문자의 특성 및 행동에 대한 자료를 의미합니다. 본 서비스는 이용자가 본 약관에 포함된 조건 및 고지 내용을 수정 없이 수락할 것을 조건으로 이용자에게 제공됩니다.

1. 용어 정의 "계정"은 본 서비스를 위한 청구 계정을 의미합니다. 하나의 사이트에 링크된 모든 프로필 (Profile)의 페이지 뷰 (Page View)는 해당 사이트에 대한 서비스 요금을 결정하기 전에 합산됩니다. "고객 자료"는 UTM을 사용하여 수집되어 본 서비스에 전달되고 프로세스 소프트웨어에 의해 분석되는 웹사이트 방문자의 특성 및 행동에 대한 자료를 의미합니다.

"과적 무서"는 오가이 등에 이 미스할 수 있는 무서로 비록하며 프로세스 소프트웨어와 함께 사용할 수 있도록 Google이 이용자에게 할

이용약관 ☒ 예, 이 이용약관에 동의합니다

계정 만들기 취소

추적 코드 설정

추적 상태 정보

웹 속성 이름 쿨펀치의 세상리뷰

웹사이트 URL http://coolpunch.tistory.com

웹 속성 ID

추적 상태 추적 코드가 설치됨

일반용 고급 맞춤

1. 무엇을 추적하고 있습니까?

● 단일 도메인

예: coolpunch.tistory.com

● 여러 하위 도메인을 보유한 단일 도메인

예: www.tistory.com

apps.tistory.com

store.tistory.com

● 다수의 최상위 수준 도메인

예: coolpunch.tistory.uk

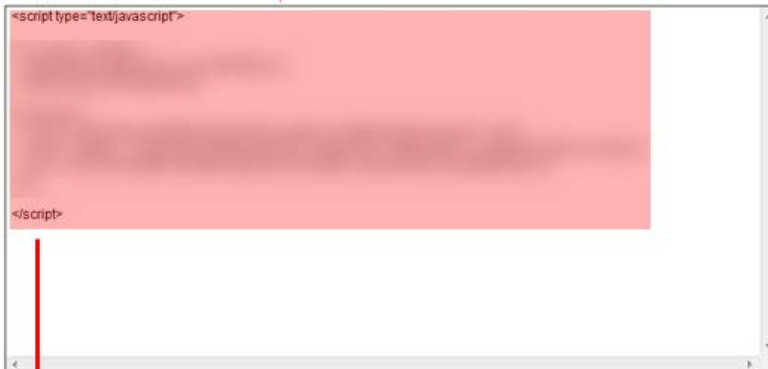
coolpunch.tistory.cn

coolpunch.tistory.fr

☐ 애드워즈 캠페인

2. 이 코드를 사이트에 붙여넣기

아래 코드를 복사한 후 추적하려는 모든 페이지에서 달기 태그인 </head> 바로 앞에 붙여넣으십시오. ?



이 도움말을 이메일로 보내기 선택사항

한 번 이 페이지에서 새 추적 코드와 기존 추적 코드를 둘 다 사용하면 보고서에 부정확한 데이터가 기록될 수 있으므로 권장하지 않습니다. 단, 사이트의 일부 페이지에는 새 추적 코드를 사용하고 나머지 페이지에는 기존 추적 코드를 사용하는 것은 가능합니다.

저장 취소

```

<!-- 구글 애널리틱스 시작 -->
<script type="text/javascript">

  var _gaq = _gaq || [];
  _gaq.push(['_setAccount', '']);
  _gaq.push(['_trackPageview']);

  (function() {
    var ga = document.createElement('script'); ga.type = 'text/javascript'; ga.async = true;
    ga.src = ('https:' == document.location.protocol ? 'https://ssl' : 'http://www') + '.google-
analytics.com/ga.js';
    var s = document.getElementsByTagName('script')[0]; s.parentNode.insertBefore(ga, s);
  })();

</script>
<!-- 구글 애널리틱스 종료 -->

```

Copy the blurred above and paste into gConfig.Analytics.gald of config.js of the player.

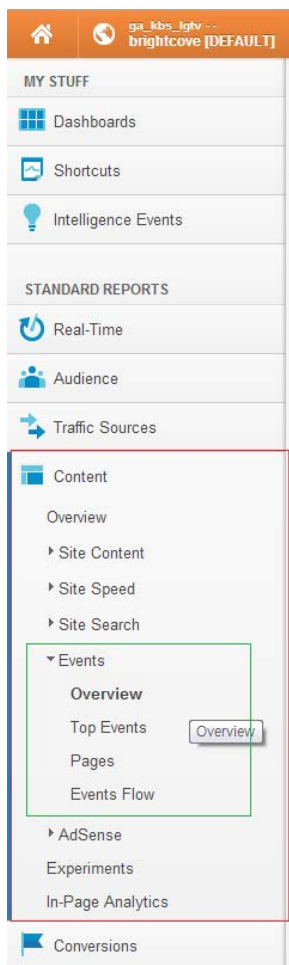
```

}» Analytics: {
  »   » gaId: "UA-35744930-1"
  » }
};

} // GA log ///////////////////////////////////////////////////
  » var _gaq = _gaq || [];
  » _gaq.push(['_setAccount', gConfig.Analytics.gaId]);
  » // _gaq.push(['_trackPageview']);
} (function() {
  »   » var ga = document.createElement('script'); ga.type = 'text/javascript'; ga.async = true;
  »   » ga.src = ('https:' == document.location.protocol ? 'https://ssl' : 'http://www') + '.google-analytics.com/ga.js';
  »   » var s = document.getElementsByTagName('script')[0]; s.parentNode.insertBefore(ga, s);
  » })();
  ///////////////////////////////////////////////////

```

Check (<http://www.google.com/analytics/>) in the Google Analytics site to see whether the web log is installed appropriately.



Click Events to see the event statistics.

Event Category	Total Events	% Total Events
1. 상품비디오 재생 통계	1,279	66.79%
2. 상품주소별 클릭 통계	318	16.61%
3. 쿠폰인트별 클릭 통계	318	16.61%

[view full report](#)

2) Other

_gaq is Google Analytics API. You do not need to modify the file.

gConfig.Category.func is a set of category related functions. This is the function for handling the above settings internally that it does not need to be modified.

3.2 CSS Environment Configuration

3.2.1 CSS Admin Environment Configuration

* Edit video Admin.css file to change configuration.

- ① Change Logo: Edit .logo {}. Edit width (logo image width), height (logo image height), and background (logo image URL)

Must be edited together as they interact with .logoBox{} and .headerBG1{}

[For Logo Image of videoAdmin.css]

```
.logoBox {  
  
    display:table;  
  
    width:1100px; height:100%;  
  
    margin-left:auto; margin-right:auto;  
  
    text-align:left;  
  
}  
  
.logo {  
  
    width:159px; height:38px;  
  
    margin-top:17px;  
  
    background:url(../images/logo-corporate-new.png)  
  
        no-repeat top left;  
  
    z-index:2;  
  
}
```

- ② The number of items to be displayed on the page can be changed with the Select tag of index.html, and default setting is configured in config.js.

- ③ Change Font: Modify the font-family of * {}.

```
* {  
  
    position:relative;  
  
    font-family: '맑은고딕', '돋움',arial,Helvetica,sans-serif;  
  
    font-size:11px;  
  
}
```

- ④ Change Language: Modify gLanguage in config.js.

3.2.2 CSS Player Environment Configuration

Change configuration by modifying player.css file.

- ① Change Font: Modify font-family of * {}.
- ② Change Logo: Edit .logo {} and edit width (logo image width) and height (logo image height)
background image can be edited in config.js.

Must be edited together as they interact with .logoBox{} and .Header.
- ③ Change Language: Modify Language Code of config.js.

4. Role of File and Function List

4.1 Admin

- videoAdmin.js: Execute script for overall Admin.
- config.js: Related with other modifiable matters and Brightcove API related matters.
- jjfw.util.js: Executes other utility wise functions
- videoAdmin.css: Overall style of Admin.

[Files inside apifolder are written, read api related sources]

Function/Variable Name	Description
_O.Vars { }	Variables and functions used by Admin.
bbsClassName []	Name of class used by Main
bbsTitleName []	Title Text to be used by Main.
eventClassName []	Name of class to set event.
mainBBSName	Variable for identifying Main and Cue Point Popup.
stripslashes()	Function for erasing backslash, if there is any.
getAttribute()	Function for obtaining attributes of the tag.
getCuepointList()	Sort function of cue point. Returns the product text of the cue point where type is 1
getModifyDate()	Returns final date of modification
_O.init()	Initializes overall Admin variables.
_O.setEvent()	Sets event for main screen
_O.CheckStrLen()	Checks whether the maximum number of characters in the TextArea has been exceeded
_O.List { }	A set of functions and variables related with items and page listing
getTotalPageNum()	Function for returning the total number of pages listed.
setOrderByOption()	Function for setting the value for sorting while listing.
getOrderByOption()	Function for sorting based on the value set by setOrderByOption.
getReadUrl()	Return feature of read API.
getRefreshReadUrl()	read API return forrefreshing
getWriteUrl()	write API return.
refreshItem()	Function for refreshing listed items.
loadList()	Function that calls getReadUrl with parameter values.
loadList2()	Function that calls getRefreshReadUrl with parameter value ID when using write API.
onRefreshItem()	Function for calling the function which displays the items received when the screen is refreshed.
onloadList()	Sets the items loaded by read API to variables, displays them on the main screen, and divides them into pages
writeltemTag()	Function for writing, modifying, and deleting values according to the type of the parameters received.
search()	Function that searches with the input value.

pageMove()	Function for page navigation.
checkPage()	Function for checking current page when navigating through pages.
setAscDesc()	Function for returning DIV suitable for the value in order to sort according to the value set in setOrderByOption.
init()	Initializes variables required for listing.
orderByIdx	Variable for sorting the list.
haveMetadata[]	Storage space for metadata.
oderList()	Calls loadList() after sorting
_O.start { }	Function for starting the script
_O.setMainHtml()	Function for setting the main screen and setting the event.
_O.setBBSHtml()	Function for setting main screen and Cue Point popup HTML
_O.setBBSContentHtml()	Function for setting main screen and Cue Point popup HTML
_O.setPageHtml()	Function for configuring page div.
_O.Player { }	A set of variables and functions related with the player.
Vars{ }	A set of variables required by player.
init()	Function for initializing variables used by the player.
smallPlayerLoaded()	Function loaded when the player is loaded. Loads modules and events.
smallPlayerProgress()	Function for the event always called when the player is started.
setSmallPlayer()	Function for setting player HTML.
smallPlayerEvent()	Function for setting the event.
_O.Dialog { }	A set of variables and functions related with popup.
detailInfoText	Cue Pointpopup text variable
detailInfoClsName	DIV class name used for Cue Pointpopup.
detailBBSTitleName	Title text for cue point popup list
detailBBSTitleClsName	Cue Pointpopup list class name
detailBBSName	Variable for differentiating Main and Cue Point popup
init()	Initializes Dialog related variables
setEvent()	Sets DIV event in the Dialog.
setAllDialog()	Declares all popups used in Admin.
setNewRegisterDialogHtml()	Newly registered popup HTML.
setDetailInfoDialogHtml()	Cue point popup HTML
useTagClick()	Function related with tags that can be used in cue point popup.
taggedTagClick()	Function related with tags used in cue point popup.
setDetailBBSHtml()	Cue point popupHTML
setCuePointSettingHtml()	Cue point popup HTML
setPublishHtml()	Pub.Code popup HTML
setCuePointPreviewPlayerHtml()	Preview in cue point popup, watch video popup HTML
dialogClick()	Function related with click event.
dialogOver()	Mouse over related function
dialogOut()	Mouse out related function
cuePointAddSaveBtnClick()	Popup HTML for confirming the addition and modification of cue point

4.2 Player

- api.js : Loads JavaScript and pastes it inside head tag. Calls ctMain.start when the loading ends.
- config.js : Other changeable matters and Brightcove API related matters and Analytics related matters.
- ctAnalytics.js : Matters related with Google Analytics.
- ctCuePoint.js : Cue Point UI and player related
- ctLanguage.js : Matters related with language according to country code.
- ctList.js : Matters related with listing.
- ctMain.js : Overall settings for player and listing.
- ctPlayer.js : Matters related with player

- ctStyle.js : Matters related with style according to cue point type.
- jjfw.util.js: Executes other utility features
- player.css : Overall style of the player.

1) ctMain.js

_O.Vars { }	A set of variables used in ctMain
param { }	Parameter value.
apiBaseUrl	Variable to save Read API
_O.reload()	Reloaded function.
_O.start()	Start function of ctMain.
_O.URL { }	Read API related.
getSearchUrl ()	Search related read API
getFindUrl ()	Video find related read API
_O.Html { }	HTML setting related
set ()	HTML setting
getHeader ()	Header HTML
getSearchBox ()	SearchBox HTML
getPlayerBox()	PlayerBox HTML
getCtListBox()	ListBox HTML
getFooter ()	Footer HTML
_O.Search { }	Searching
setQuery()	Sets the string to search
getQuery()	Gets the string searched
go ()	Reloads with the searched word.
setEvent ()	Event setting

2) ctPlayer.js

_O.start()	Start function of ctPlayer.
_O.Data { }	Item related.
id	ID of loaded item
item	Variable to save loaded item.
load ()	Load item using readAPI.
onLoad()	Sets HTML, sets event, sets style, and initializes analytics based on loaded items after the loading completes
_O.InfoSNS { }	SNS and video information related
setHtml ()	infoSNS HTML Set.
getTwitterHtml ()	HTML for Twitter share button
getFacebookHtml ()	HTML for Facebook share button
_O.Player { }	Player related
Vars { }	A set of variables used by the player
init()	Initializes player variable.
play ()	Plays with player.
pause ()	Pauses player.
seek ()	Seeks with player.
commercePlayerLoaded ()	Function called after the player is loaded.
commercePlayerReady ()	Function called when the player is ready.
commercePlayerEvent()	Function executed when a certain player event occurs.
commercePlayerProgress()	Function always called when the player is started.
setCommercePlayer()	Sets player HTML.
_O.onLoadBrightCove()	Creates player object.
_O.Event{ }	Related with events other than player events
set()	Sets event.

3) ctList.js

_O.Vars { }	A set of variables
initialPlay	Set to 0 when there is Video ID, or 1 if there is no Video ID or it is undefined
jParam { }	Variable to save the parameter to.
_O.start()	The first function to start on the ctList.
_O.Item { }	Related with 1 item only
id	Variable to save the Video Item ID to.
load()	Loads 1 item using Read API.
_O.Paging{ }	Paging related
page_number	Page number variable.
page_size	Number of items to show.
total_count	Number of all items.
set()	Sets variables to values.
getTotalPageNum()	Returns the total number of pages.
go()	Moves to a page.
setHtml()	Sets page HTML.
setEvent()	Sets paging and page navigation related events.
_O.Category{ }	Related with the category of the list.
ctIdx	Category index
go()	Moves to a category.
setHtml()	Sets category HTML.
setEvent()	Sets category related events.
_O.List { }	Related with listing
items[]	Variable to save loaded items to.
curlId	ID of current video
load()	Loads items using Read API
onLoadItem()	Sets the variable to a loaded item.
onLoad()	Sets variables to loaded items and sets HTML.
go()	Moves to a corresponding item.
setHtml()	Sets list HTML.
setEvent()	Sets main screen related events.

4) ctCuePoint.js

_O.Metadata { }	Related with the metadata loaded
length	Total length of metadata
index { }	Metadata is made of an array, and this declares what the index of the array indicates.
_O["CueTypeA"] { }	Related with Cue Point Type A
points[]	Variable to save the cue point to.
Idx	Index of the array to which the cue point is saved.
openWindow()	Executed when overlaying or opening a new window.
closeWindow()	Executed when closing a window
getPoints()	Gets cue point.
setPoints()	Sets and sorts cue points
Animation{ }	Related with cue point animation
init()	Initializes cue point animation
activeItem()	Changes cue point position using Attribute.
inActiveItem()	Changes cue point position using Attribute.
refresh()	Cue Point refresh
open()	Cue Point UI Open
progressShow()	Called continuously when player's progress event occurs. Function for checking and showing cue points.
refreshMediaPlay()	Called when the play event of the player occurs.
refreshMediaSeek()	Called when the seek event of the player occurs.
refreshMediaComplete()	Called when the complete event of the player occurs.
Event{ }	Cue point related event.

onOver()	Cue point related mouse over event.
onOut()	Cue point related mouse out event.
onClick()	Cue point related mouse click event.
set()	Cue point related event setting.
Arrow { }	Related with cue point up down arrow
init()	Cue point arrow initializing function.
onOver()	Cue point arrow mouse over event function
onOut()	Cue point arrow mouse out event function
onClick()	Cue point arrow mouse click event function
checkOnOver()	Function for checking cue point arrow focus.
Scroll{ }	Related with cue point scroll
move()	Function for moving the scroll.
isScrolling()	Checks scrolling
markStart()	Starts mark scrolling.
markStart()	Ends mark scrolling.
setHtml()	Sets cue point HTML.
setCtStyle()	Apply cue point related style.
getPlayerObject()	Loads player object.

5. Google Closure Compiler

Google Closure Compiler is used to execute Java script sources more quickly and obfuscates sources to prevent unwanted execution through the modification of the sources.

Please refer: <https://developers.google.com/closure/compiler/docs/api-tutorial3>

Admin Compiler

```
=====

#!/bin/sh

# https://developers.google.com/closure/compiler/docs/api-tutorial3
echo
echo
echo "J Source Compile"
echo

echo "Compile:: ./admin_working/js/*.js> ./admin_working/js/brObj.js"
/usr/local/jdk1.6.0_29/bin/java \
    -jar /home/shdev/docs/http-www.jnjt.co.kr/prj/jCompiler/compiler.jar \
    --compilation_level ADVANCED_OPTIMIZATIONS \
    --jsadmin_working/js/jjfw.util.js \
    --jsadmin_working/js/videoAdmin.js \
    --externs admin_working/js/_extern_profile.js \
    --js_output_fileadmin_working/js/brObj.js

echo "cp ./admin_working/* ./amdin_home/* except admin.html"
rm -rf ./admin_home/api
rm -rf ./admin_home/css
rm -rf ./admin_home/images
rm -rf ./admin_home/js
mkdir ./admin_home/js
cp ./admin_working/js/config.js ./admin_home/js/
cp -r ./admin_working/js/jquery-UI ./admin_home/js/
cp -r ./admin_working/css ./admin_home/
cp -r ./admin_working/images ./admin_home/
cp -r ./admin_working/api ./admin_home/
cat ./copyright.txt ./admin_working/js/brObj.js > ./admin_home/js/brObj.js
echo
echo "Done !!"
echo
```

=====

ADVANCED_OPTIMIZATIONS is used for the compile level. This also subjects the case of JSON function to compiling.

externs is for designating a variable within the Java Script that must not be replaced or a function, and it is defined in _extern_profile.js in jsfolder.

For the folder structure, admin_working is used as the source folder and compiled sources are saved to admin_home.

player_working and player_home are used also for Player.

```
=====

#!/bin/sh

# https://developers.google.com/closure/compiler/docs/api-tutorial3
echo
echo
echo "J Source Compile"
echo

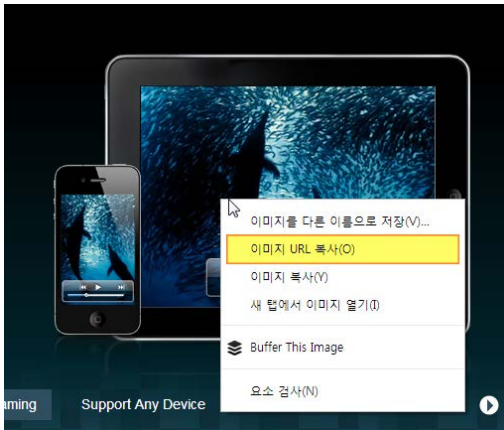
echo "Compile:: ./player_working/js/*.js> ./player_working/js/brObj.js"
/usr/local/jdk1.6.0_29/bin/java \
    -jar /home/shdev/docs/http-www.jnjt.co.kr/prj/jCompiler/compiler.jar \
    --compilation_level ADVANCED_OPTIMIZATIONS \
    --jsplayer_working/js/jjfw.util.js \
    --jsplayer_working/js/ctList.js \
    --jsplayer_working/js/ctCuePoint.js \
    --jsplayer_working/js/ctPlayer.js \
    --jsplayer_working/js/ctMain.js \
    --jsplayer_working/js/ctAnalytics.js \
    --externs player_working/js/_extern_profile.js \
    --js_output_fileplayer_working/js/brObj.js

echo "cp ./player_working/* ./player_home/* except player.html and js/api.js"
rm -rf ./player_home/css
rm -rf ./player_home/images
cp ./player_working/js/config.js ./player_home/js/
cp ./player_working/js/ctStyle.js ./player_home/js/
cp ./player_working/js/ctLanguage.js ./player_home/js/
cp ./player_working/js/jquery*.js ./player_home/js/
cat ./copyright.txt ./player_working/js/brObj.js > ./player_home/js/brObj.js
cp -r ./player_working/css ./player_home/
cp -r ./player_working/images ./player_home/

echo
echo "Done !!"
echo
=====
```


6. FAQ

Q.1. How can I load the thumbnail image URL used for Cue Point?



: Commerce video template is built to load the image that is already posted to your web page to use it as a thumbnail.

For image URL, right click on the image on the web and copy the image to the clip board by selecting Copy Image URL and then paste it to the Commerce Template Thumbnail URL. Click the 'View Thumbnail' button to preview the image.

Q.2. Why does it take 3-5 minutes to apply the settings I made to the template?

: It is the time required to synchronize Media API. It can take minutes to automatically save and apply the metadata you set in the commerce video template to Video Cloud Studio, and the time can vary depending on the network situation. Update the setting and wait until the information you modified or set to commerce video template is applied so that the metadata of the cue point is saved without loss.

Q.3. I want to use other types of cue points other than those set in the template by default, how can I add them?

: If you wish to use other types of cue points in addition to the default cue point setting and animations, then you need to develop them additionally. Please contact the Brightcove Korea Team for details.

Q.4. Can I use the clickable videos with the Video Cloud Express version?

: You can use commerce video template from the version that supports Read API and WRITE API. Read API and WRITE API are supported from PRO version. Click [here](#) about the functions supported by each version of the Video Cloud.

Q.5. Can I play the clickable video in my mobile device?

: Commerce video template was developed with HTML5 so it can be played on a PC as well as a number of mobile devices – Apple iPad, Android tablet, Android Mobile phone, etc. However, iPhone only supports full screen video play due to Apple's policy that you can only play it as a video without clickable cue points.