

# Index

- **Building creative via Adobe Animate CC**
  - [Case 1 : Build from scratch](#)
  - [Case 2 : If the creative already built to other vendor's spec, how to update the code to DFP spec](#)
- **Building creative via Google Web Designer**
  - [Case 1 : Build from scratch](#)
  - [Case 2 : If the creative already built to other vendor's spec](#)
- [Generic HTML clickTag guidelines](#)
- [HTML5 creatives must be SSL compatible](#)

# Content

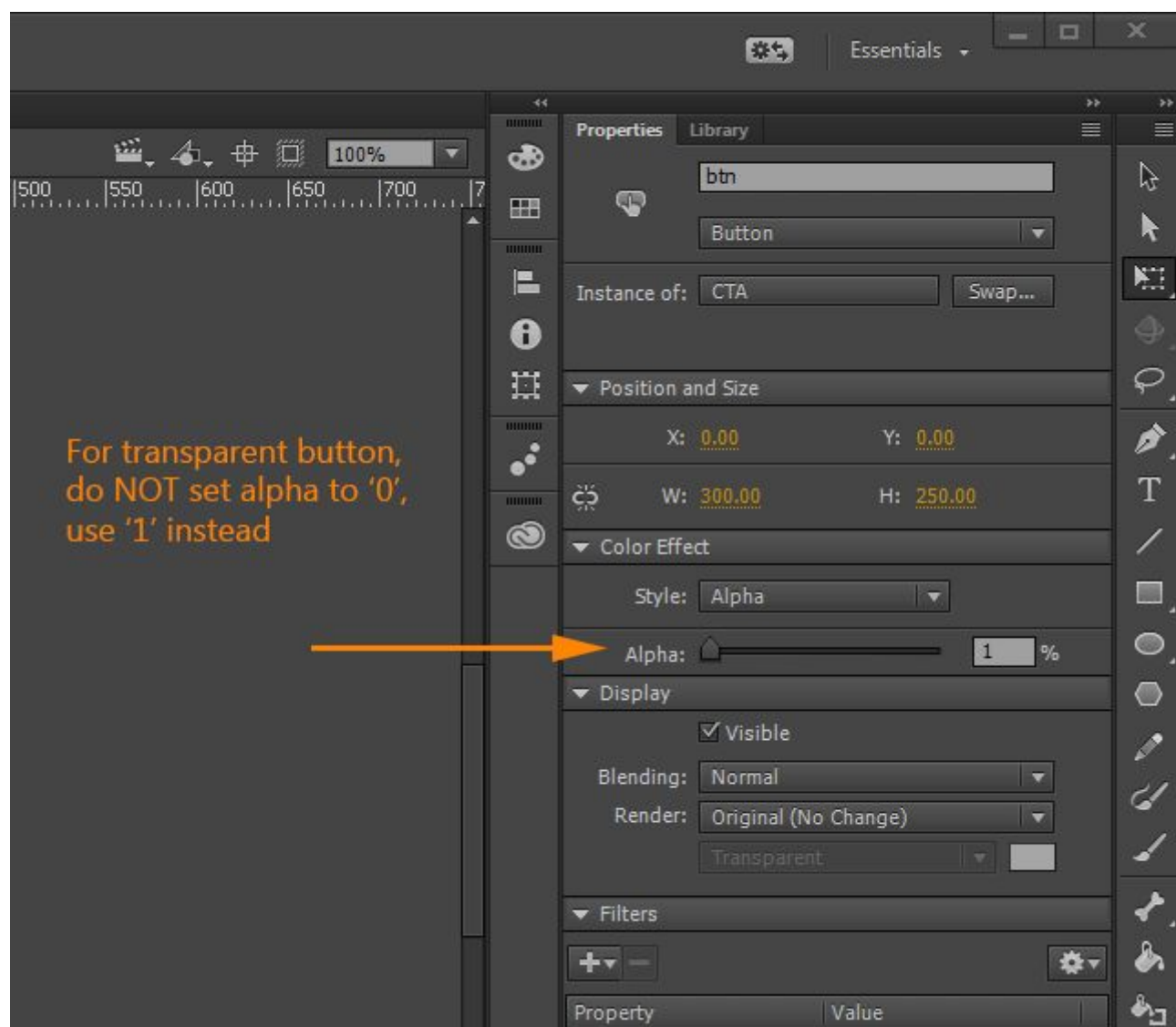
## Building creative via Adobe Animate CC

### Case 1: Build from scratch

#### Step 1:

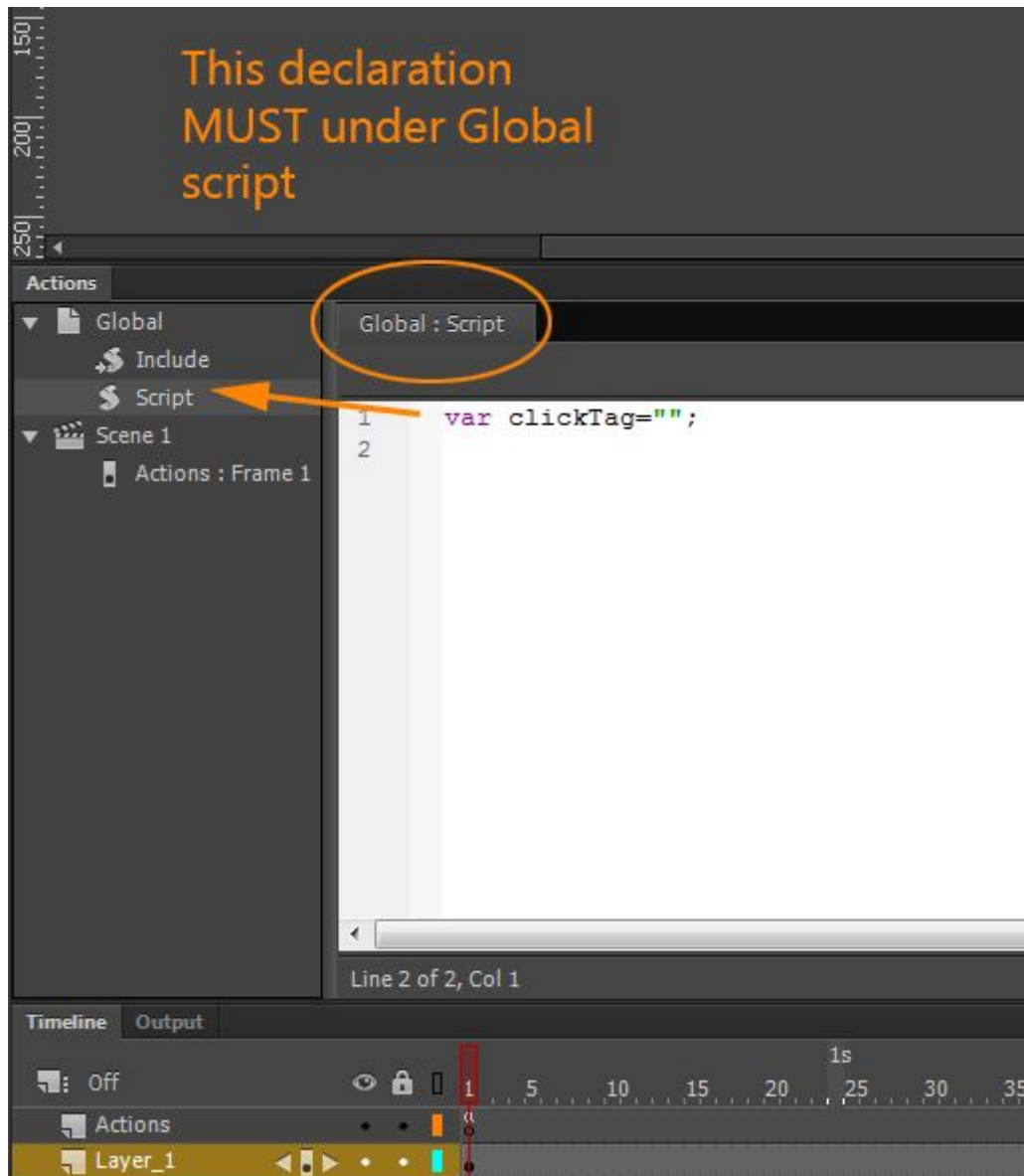
Create a button on the stage(canvas), and name it “btn”.

For transparent button, DO NOT set the alpha value to 0%, set it to 1% (see screenshot below)



## Step 2:

Declare “clickTag” variable under “Global : Script”



### Step 3:

Add event listener and function for your click button.

You can activate the click button any time on your timeline when needed. Simply create a key frame, then add the script on that frame.

The image shows a software interface with a script editor and a timeline. The script editor contains the following code:

```
1  this.btn.addEventListener("click", clickThrough);  
2  
3  function clickThrough() {  
4      window.open(clickTag, "_blank");  
5  }  
6
```

Annotations in orange text provide instructions:

- "The object's name in your script need to match the instance name on the stage" points to the `btn` property in the script.
- "You can choose when the click button activated on your timeline" points to the "Current frame" dropdown menu in the Actions panel.

The interface also shows a timeline at the bottom with a keyframe marker at frame 1.

## Case 2 : If the creative already built to other vendor's spec, how to update the code to DFP spec

### Step 1:

Declare "clickTag" variable in the header of your HTML document

```
<script type="text/javascript">var clickTag = "";</script>
```

```
index.html x
1 <!DOCTYPE html>
2 <!--
3     NOTES:
4     1. All tokens are represented by '$' sign in the template.
5     2. You can write your code only wherever mentioned.
6     3. All occurrences of existing tokens will be replaced by their appropriate values.
7     4. Blank lines will be removed automatically.
8     5. Remove unnecessary comments before creating your template.
9 -->
10 <html>
11 <head>
12 <meta charset="UTF-8">
13 <meta name="authoring-tool" content="Adobe_Animate_CC">
14 <meta name="ad.size" content="width=728,height=90">
15 <title>index</title>
16 <script type="text/javascript">
17     var clickTag = "";
18 </script>
19 <!-- write your code here -->
20 <script src="script.js"></script>
```

### Step 2:

Wrap the canvas tag with the below script

```
<a href="javascript:window.open(window.clickTag)">
```

```
<!-- Your canvas goes here --></a>
```

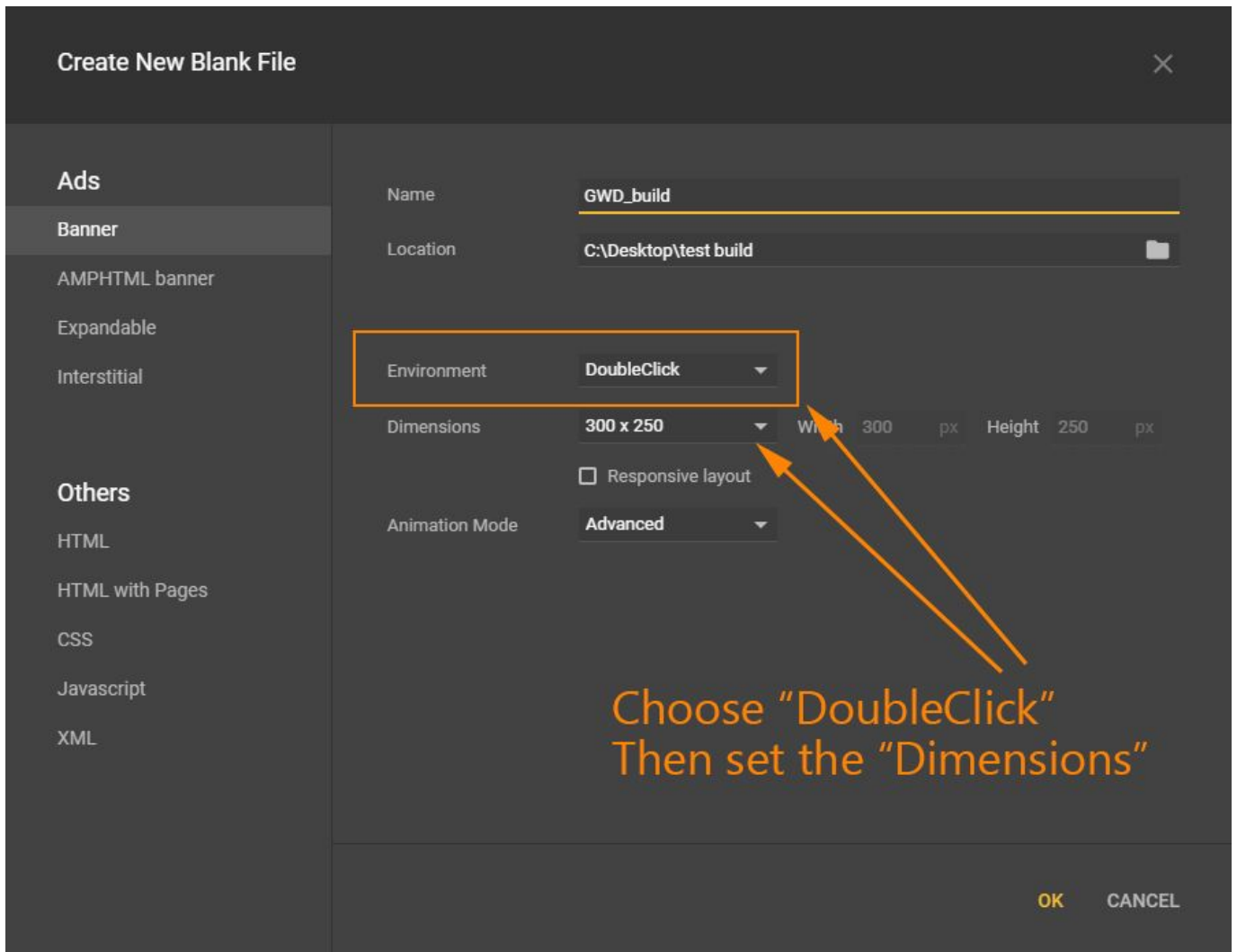
```
274 <body onload="init();" style="margin:0px;">
275 <div id="animation_container" style="background-color:rgba(94, 179, 58, 1.00); width:728px; height:90px">
276 <a href="javascript:window.open(window.clickTag)"><canvas id="canvas" width="728" height="90" style="
277     position: absolute; display: block; background-color:rgba(94, 179, 58, 1.00);"></canvas></a>
278 <div id="dom_overlay_container" style="pointer-events:none; overflow:hidden; width:728px; height:90px;
279     position: absolute; left: 0px; top: 0px; display: block;">
280 </div>
281 </body>
282 </html>
```

# Building creative via Google Web Designer

## Case 1 : Build from scratch

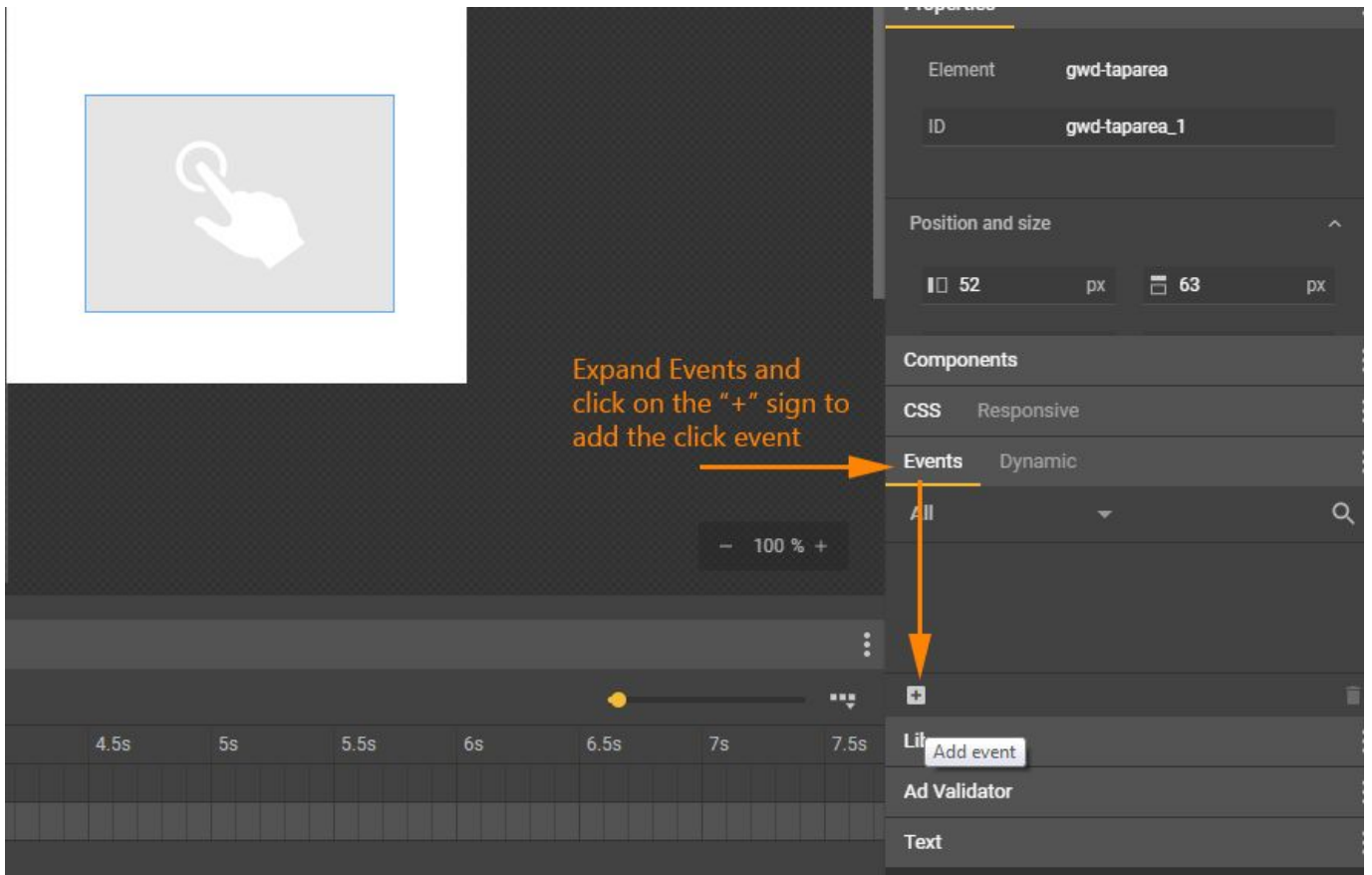
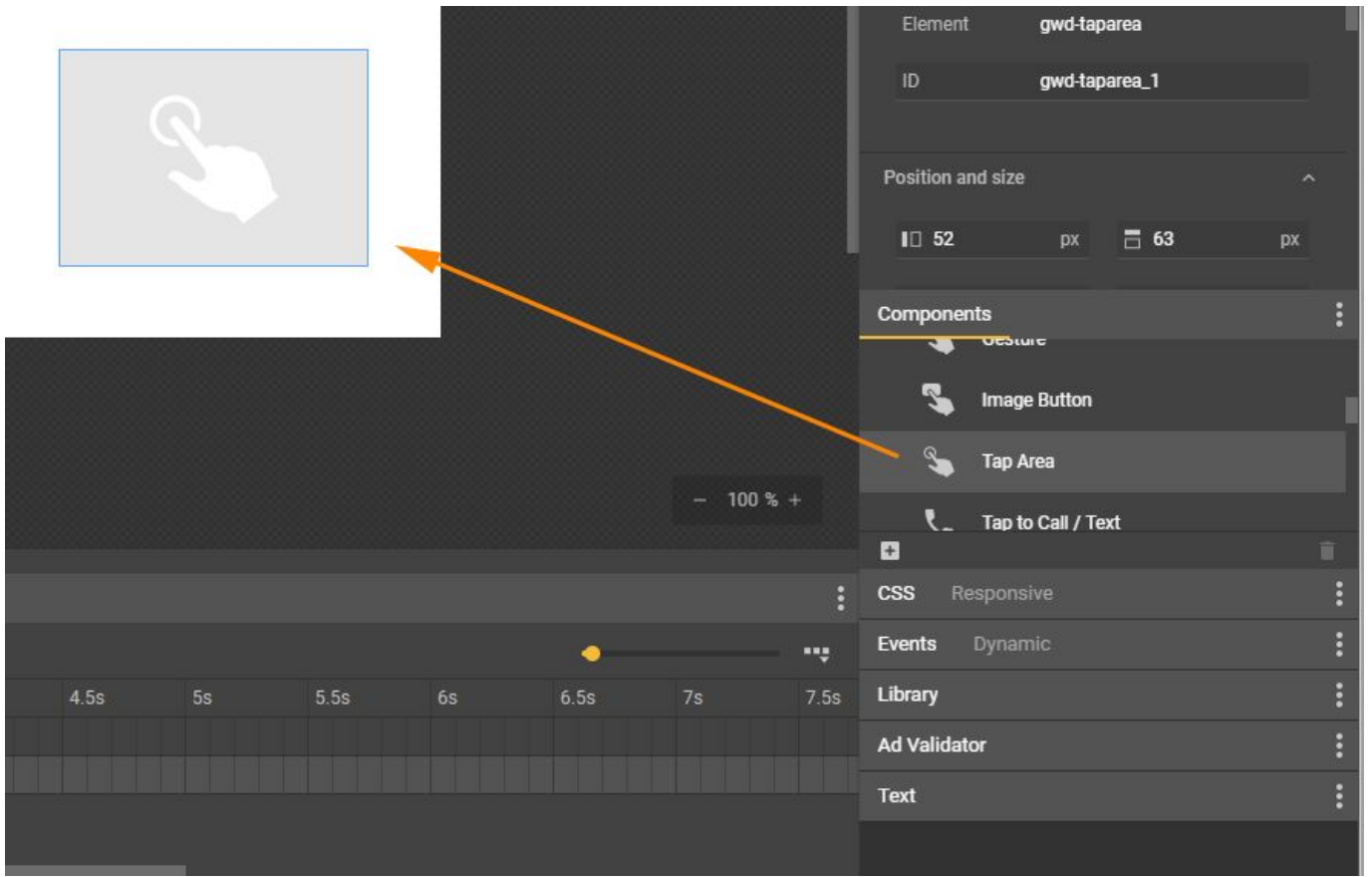
### Step 1:

Choose “DoubleClick” as the “Environment” when start building the creative



### Step 2:

Drag a Tap Area component onto stage(canvas), then expand Events tab, add the event for click through



gwd-taparea\_1   event   action   receiver   configuration

### Event

- Mouse
- Touch
- Tap Area

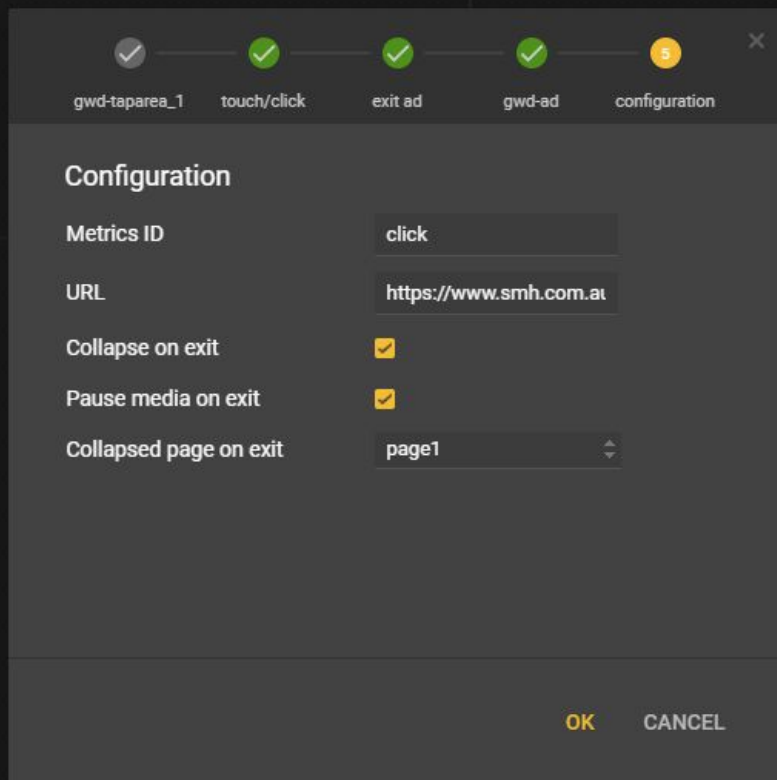
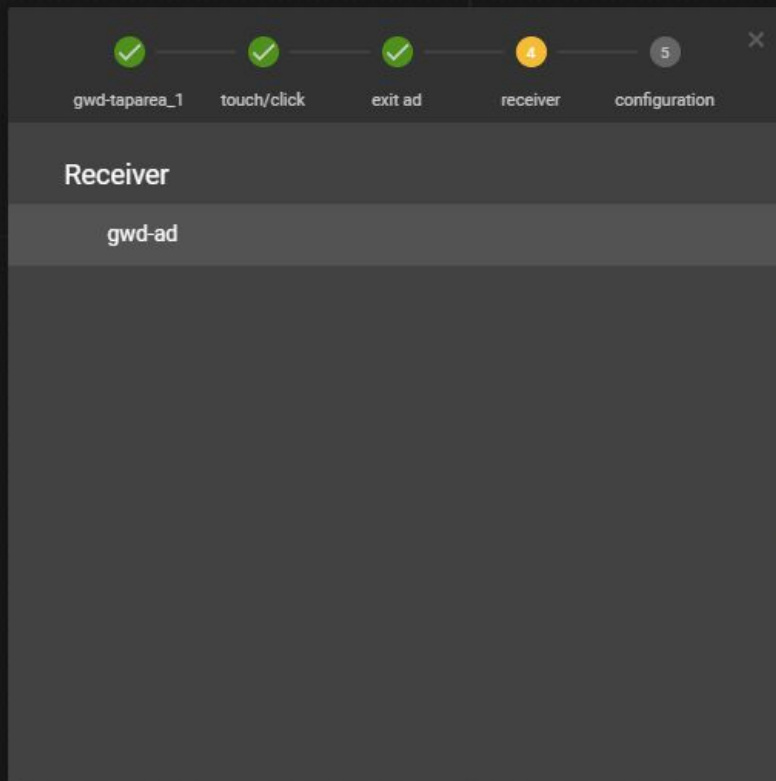
→ Touch/Click

gwd-taparea\_1   touch/click   action   receiver   configuration

### Action

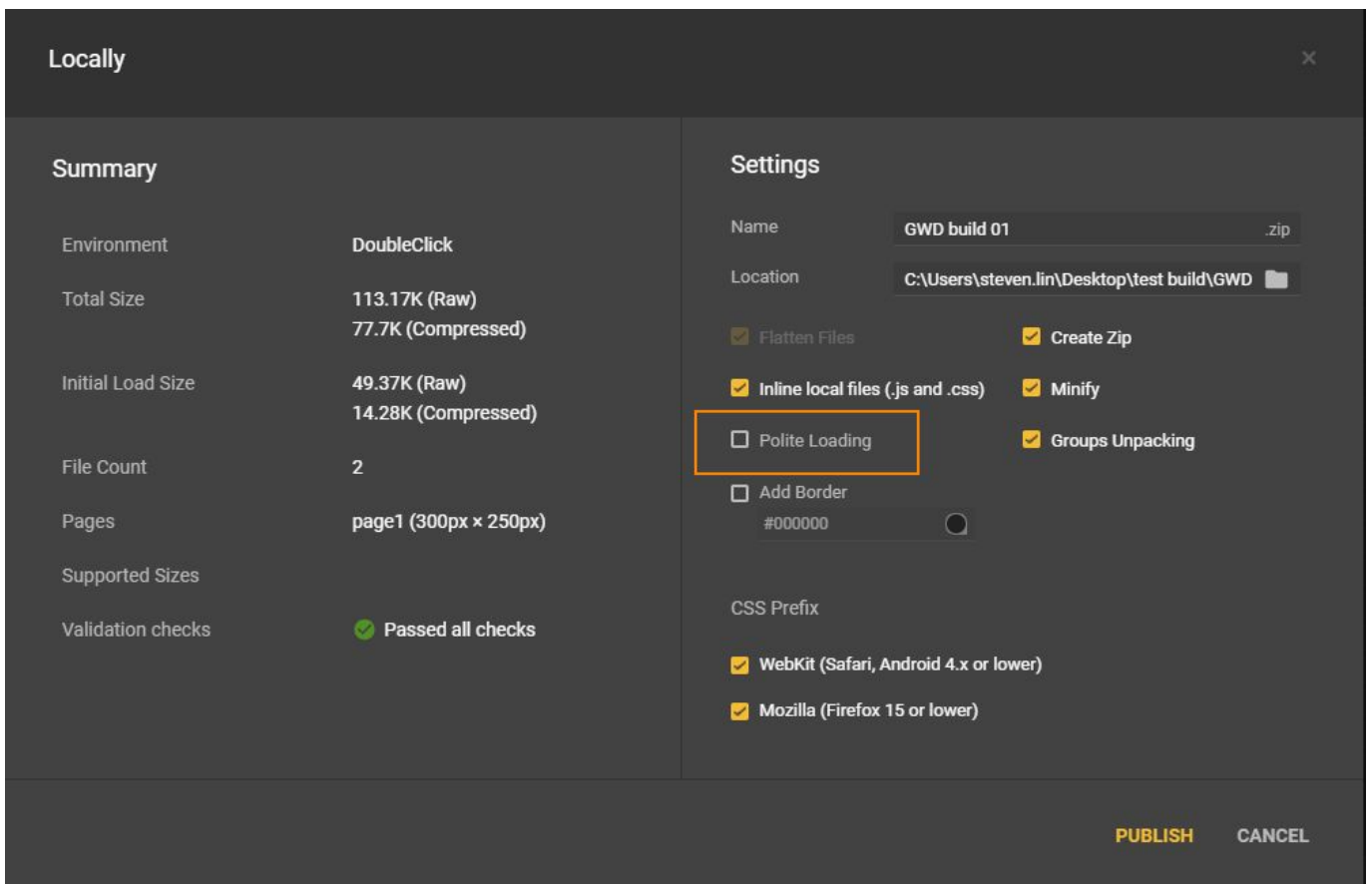
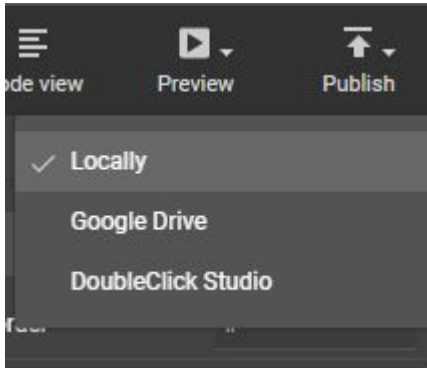
- CSS
- Google Ad
- Initialize ad
- Go to page
- Exit ad
- Exit ad (overrides URL)
- Increment counter
- Start timer
- Stop timer





### Step 3:

When publish the creative, choose “Locally” in Publish options, and GWD will give you a summary of the file weight, initial load size etc. If the initial load is too big, tick the “Polite Loading” box so the creative will be loaded after page content loaded



## Case 2 : If the creative already built to other vendor's spec

Check the HTML code of the index.html, search for keyword "environment". If you see "gwd-doubleclick", then go to step 2, if it's anything else, send the creative back and let the agency/client amend their creative, because DFP **ONLY** accept GWD creative with "DoubleClick" environment, other than that, DFP will just reject it regardless if you have the correct "clickTag" script or not

```
1 <!DOCTYPE html>
2 <html><head><meta name="GCD" content="YTk30DQ3ZWZhN2I4NzZmMzBkNTEwYjJlF7405ee94886a448ad7d4f0f6497d5c1"/>
3 <meta charset="utf-8">
4 <meta name="generator" content="Google Web Designer 3.0.1.0319">
5 <meta name="template" content="Banner 3.0.0">
6 <meta name="environment" content="gwd-doubleclick">
7 <meta name="viewport" content="width=device-width, initial-scale=1.0">
```

### Step 2:

Search the keyword "gwd.actions.gwdDoubleClick.exit" in the HTML code, and you will find the code block of click/tap event handler, something like the below script

```
<script type="text/javascript">
```

```
gwd-events="handlers">gwd.auto_Gwd_taparea_1Action=function(event){gwd.actions.gwdDoubleClick.exit("gwd-ad","exit",true,true,"page1")}</script>
```

Replace the "exit" (including double quotation mark) with clickTag(without any quotation mark), so it will look like this:

```
<script type="text/javascript">
```

```
gwd-events="handlers">gwd.auto_Gwd_taparea_1Action=function(event){gwd.actions.gwdDoubleClick.exit("gwd-ad",clickTag,true,true,"page1")}</script>
```

Save the new index.html file and zip the creative for DFP.

If you can't find the keyword "gwd.actions.gwdDoubleClick.exit", that means the exit event is a custom setup, either send the creative back to client/agency or raise with AdTech team for investigation.

## Generic HTML clickTag guidelines

Example of a click tag inserted in an HTML document:

```
<html>
<head>
<meta name="ad.size" content="width=300,height=250">
<script type="text/javascript">
var clickTag = "http://www.google.com"; </script>
</head>
[The rest of your creative code goes here.]
</html>
```

Your creative must use the click tag variable as the click-through URL:

```
<a href="javascript:window.open(window.clickTag)">

</a>
```

## HTML5 creatives must be SSL compatible

HTML5 creatives must be SSL compatible to serve to HTTPS sites.

So all URLs to external JS, image assets, video etc must start with “https”, not “http”.