

Trade Desk Creative Guidelines

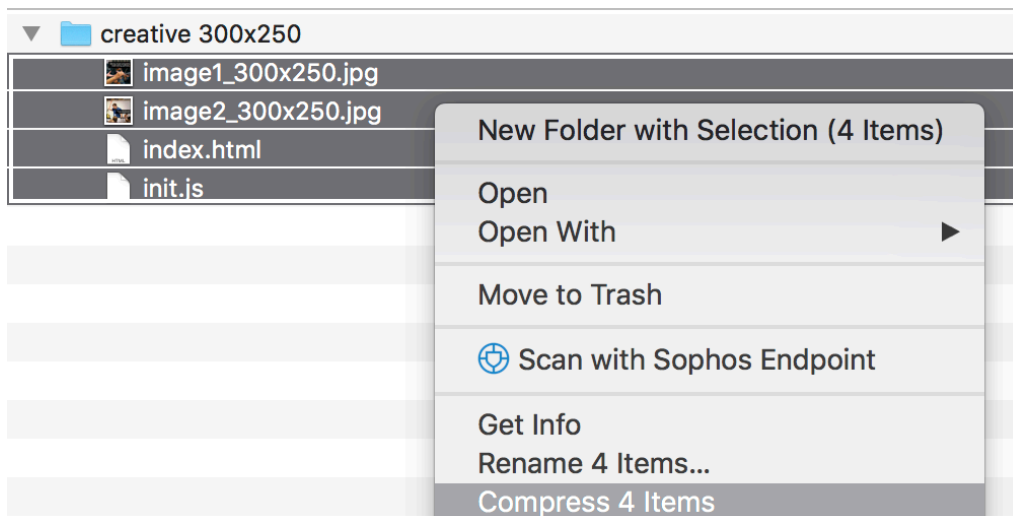
This article details best practices for creatives that will be loaded directly into The Trade Desk's platform.

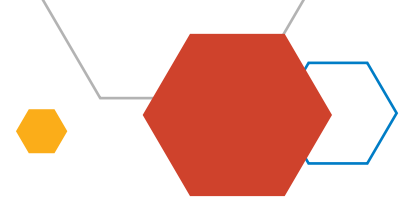
- [Hosted Ad Creative Specifications](#)
- [Adding Trade Desk Click Tags](#)
- [Formatting .zip Archives for Trade Desk Creatives](#)

Hosted Ad Creative Specifications

- All information for serving the creative (with the exception of commonly used JavaScript and CSS libraries) should be included in the HTML file.
- The maximum HTML file size is 200 kilobytes.
- All files for displaying the creative (with the exception of commonly used JavaScript and CSS libraries) should be included in the zip file. The primary HTML file should be located in the root of the zip file.
- Zip files must contain no more than 100 files, including at least one HTML file for use as the initiating file for display. We suggest keeping the file count as low as possible to minimize browser performance impact.
- The maximum size of any automatically started video should be less than 1.1 megabytes. The maximum size of any individual file is 2.2 megabytes.
- The maximum size of the primary HTML file is 100 kilobytes.
- The zip file should be smaller than 10 megabytes compressed, and the contents should be smaller than 12 megabytes uncompressed. Only file types of HTML, JS, CSS, JPG, JPEG, GIF, PNG, and SVG may be included in the zip file.

Important Note: The content of your ad must reside **in the root directory of the .zip archive**, otherwise your .zip will be rejected by The Trade Desk. In the example below, we compress only the files **inside** of the directory 'creative 300x250' **not the folder itself**.





Adding Trade Desk Click Tags

The Trade Desk's click tags are comprised of two main parts:

1. The click handler script, which should be inserted before the closing `</head>` tag in your HTML file. Simply copy and paste:

```
<!--Begin TTD Click handler-->
<script type = "text/javascript">
  function getParameterByName(name) {
    name = name.replace(/[\[\]]/, "\\[\\]");
    var regex = new RegExp("\\?&" + name + "=(^[&#]*)");
    results = regex.exec(location.search);
    return results === null ? "" : decodeURIComponent(results[1].replace(/\+/g, " "));
  }
  var clickTag = getParameterByName("clickTAG");
</script>
```

2. The anchor `<a>` tag, which wraps the content of your ad to make it clickable like any common hyperlink:

```
<!--TTD Anchor Tag-->
<a href = "javascript:window.open(window.clickTag);void(0);">
  <!-- Content that you want to click out goes here -->
</a>
```

This screenshot shows the placement of the code within an example HTML ad:

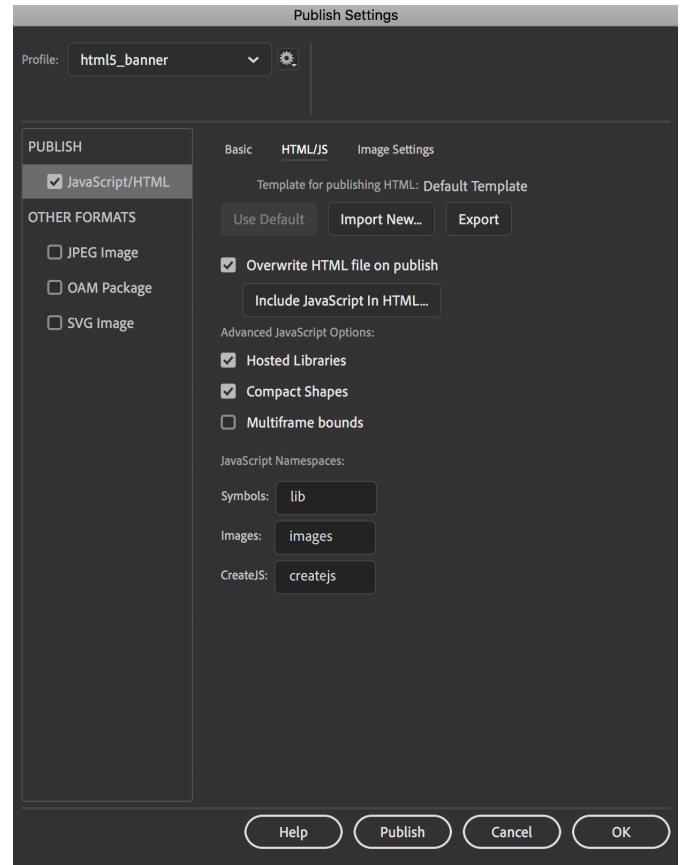
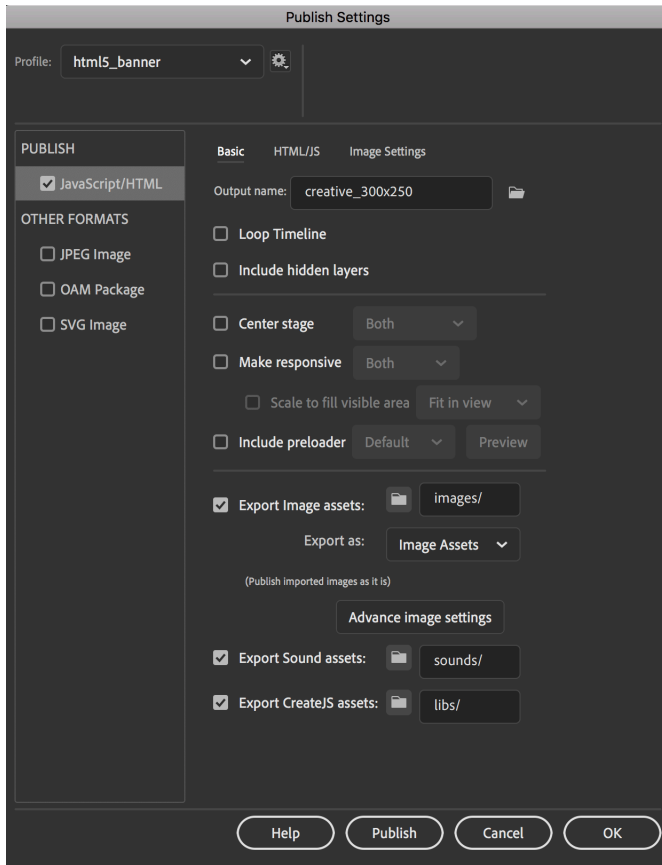
```
1 <!doctype html>
2 <html lang='en'>
3 <head>
4 <meta charset = 'UTF-8'>
5 <meta name="viewport" content="width=device-width, initial-scale=1, maximum-scale=1" />
6 <meta name="ad.size" content="width=300, height=250">
7 <script src="https://cdnjs.cloudflare.com/ajax/libs/gsap/latest/TweenMax.min.js"></script>
8 <link rel="stylesheet" type="text/css" href="style.css">
9 <script type="text/javascript" src="init.js"></script>
10 <!--TTD Click handler-->
11 <script type='text/javascript'>
12   function getParameterByName(name) {
13     name = name.replace(/[\[\]]/, "\\[\\]");
14     var regex = new RegExp("\\?&" + name + "=(^[&#]*)");
15     results = regex.exec(location.search);
16     return results === null ? "" : decodeURIComponent(results[1].replace(/\+/g, " "));
17   }
18   var clickTag = getParameterByName("clickTAG");
19 </script>
20 </head>
21 <body>
22 <!--TTD Anchor Tag-->
23 <a href="javascript:window.open(window.clickTag);void(0);">
24   <div id="container" style="visibility: hidden">
25     <div id="background"></div>
26     <div id="frame"></div>
27     <div id="text">
28       <p>Buy one, get one free!</p>
29     </div>
30   </div>
31 </a>
32 </body>
33 </html>
```





Adding TTD Click Tags to Adobe Animate Creatives

When using Adobe Animate as your HTML5 authoring tool, we recommend the following Publish settings:

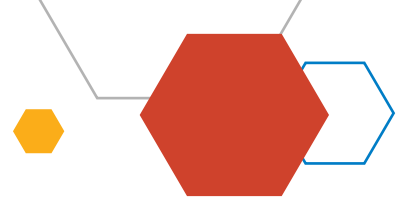


This screenshot shows the placement of the code within an example HTML ad that was published using Adobe Animate. The click handler script is added just before the closing `</head>` tag, and we wrap the entire “animation_container” `<div>` with the anchor tag so every pixel of the ad is clickable:

```

59 <!-- write your code here -->
60 <!--Start TTD Click Handler-->
61 <script type="text/javascript">
62   function getParameterByName(name) {
63     name = name.replace(/[\/\[\]\/, "\\|"/, "\\|");
64     var regex = new RegExp("[\\?&]" + name + "=[^&#]*"),
65         results = regex.exec(location.search);
66     return results === null ? "" : decodeURIComponent(results[1].replace(/\/+/g, " "));
67   }
68   var clickTag = getParameterByName("clickTAG");
69 </script>
70 </head>
71 <body onload="init();" style="margin:0px;">
72 <a href="javascript:window.open(window.clickTag);void(0);">
73 <div id="animation_container" style="background-color:rgba(255, 255, 255, 1.00); width:300px; height:600px">
74 <canvas id="canvas" width="300" height="600" style="position: absolute; display: block; background-color:rgba(255, 255, 255, 1.00);"></canvas>
75 <div id="dom_overlay_container" style="pointer-events:none; overflow:hidden; width:300px; height:600px; position: absolute; left: 0px; top: 0px; display: block;">
76 </div>
77 </div>
78 </a>
79 </body>
80 </html>

```

Formatting .zip Archives for TTD Creatives

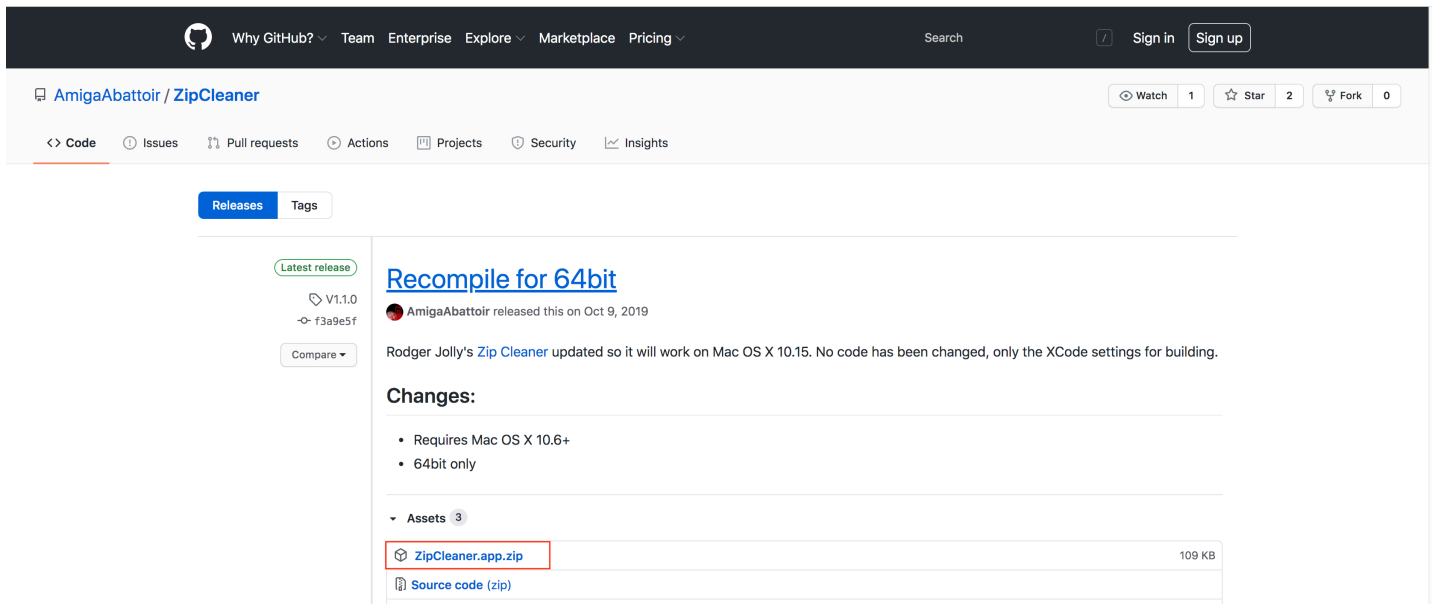
Mac OS users may run into issues when creating and uploading .zip archives to The Trade Desk's creative platform. This is due to certain hidden files that the OS includes in the archive, but are not actually needed. In the example below, the directory '_MACOSX' was created, and is hidden to users on Mac OS but appears visible in Windows 10.

Name	Date	Type	Size	Tags
📁 _MACOSX	1/5/2021 1:35 PM	File folder		
📄 Download	1/5/2021 12:32 PM	PNG File	367 KB	
📄 Preferences 1	1/5/2021 12:42 PM	PNG File	219 KB	
📄 Preferences 2	1/5/2021 12:42 PM	PNG File	114 KB	
📄 Run	1/5/2021 12:30 PM	PNG File	94 KB	
📄 Security	1/5/2021 12:33 PM	PNG File	438 KB	

The best solution to this problem is for Mac OS users to try an alternative archive application to build .zip files. Unfortunately, these applications tend to require a paid license, and don't work 100% of the time, so we'd like to share a freeware workaround to this problem.

ZipCleaner is a freeware application available on GitHub. You can download it here:

<https://github.com/AmigaAbattoir/ZipCleaner/releases/>



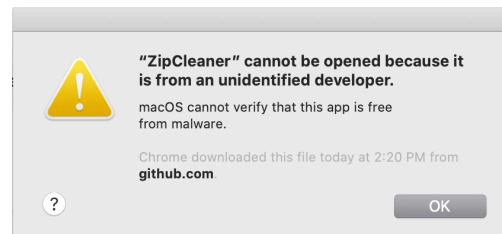
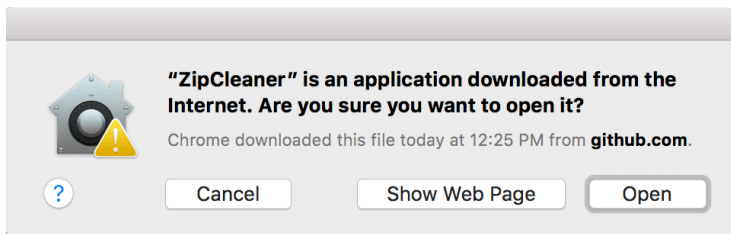
The screenshot shows the GitHub repository page for AmigaAbattoir/ZipCleaner. The page is titled "Recompile for 64bit" and was released on Oct 9, 2019. The release description states: "Rodger Jolly's Zip Cleaner updated so it will work on Mac OS X 10.15. No code has been changed, only the XCode settings for building." The "Changes" section lists: "Requires Mac OS X 10.6+" and "64bit only". Under the "Assets" section, there is a download link for "ZipCleaner.app.zip" (109 KB) and a link for "Source code (zip)".



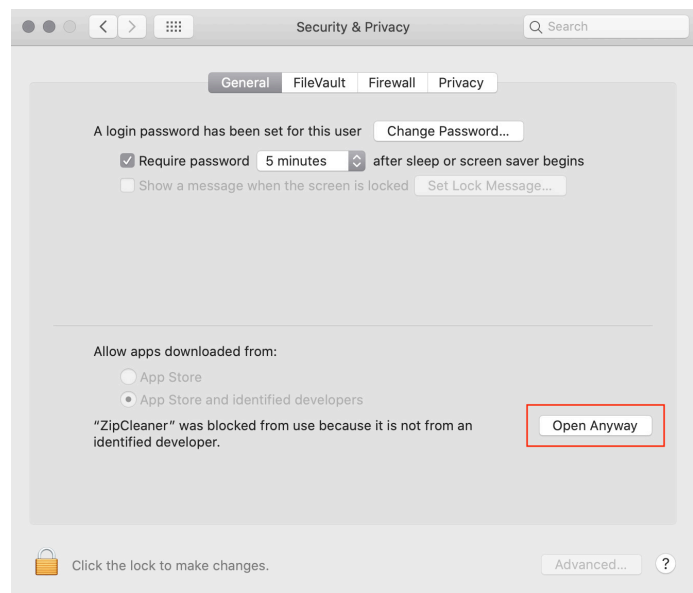
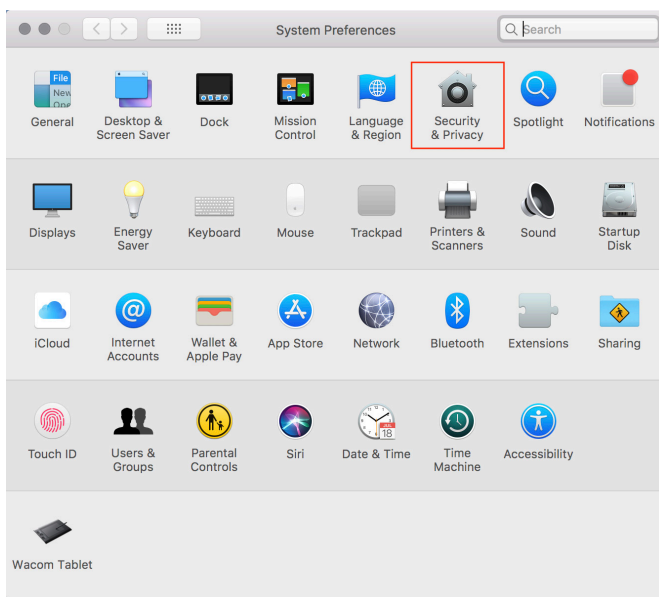


ZipCleaner Installation and Use

Download the file 'ZipCleaner.app.zip' from the link above. Extract the application file, then place it inside your user's Applications folder. When first running ZipCleaner, you will receive a prompt warning you this application was downloaded from the internet. Select 'Open'. A second prompt may alert you that it cannot be run since it wasn't downloaded from the Apple App Store. Select 'OK'.



Next, open System Preferences, select 'Security & Privacy', then select 'Open Anyway' from the lower right, and restart ZipCleaner.



When ready to use the application, simply drag and drop previously archived .zip files into the prompt to remove all extra files automatically.

