File Sizes

- Maximum initial load: 300 KB (200 KB recommended)
- Maximum subload: 600 KB
- Maximum initial file load count: 10 files
- Maximum subload file count: 100 files
- Maximum file size: 5 MB
- Maximum user-initiated load: 2.2 MB (recommended)

File-Upload Methods

The following packaging options are available for HTML5 creatives:

- A single .HTML file
- A single root/main .HTML file with supporting images
- A single root/main .HTML file with supporting images and additional supporting .JS and .CSS files.

When packaging HTML5 creatives, zip the assets, not the folder containing the assets.

Single HTML File

- 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 KB.

Zip-Compressed File

- 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 low to minimize browser performance impact.
- The maximum size of any automatically-started video should be less than 1.1 MB.
- The maximum size of any individual file is 2.2 MB.
- The maximum size of the primary .HTML file should be 100 KB.
- The .zip file should be smaller than 10 MB compressed and the contents should be smaller than 12 MB uncompressed.
- Only .HTML, .JS, .CSS, .JPG, .JPEG, .GIF, .PNG, and .SVG files may be included in the .zip file.

ClickTAG Parameters

The ClickTAG parameter is provided to the primary .HTML file as a URL parameter. The name of the URL parameter is customizable at the time of creative upload or during editing. This system is similar to how we provide click-tracking information for hosted Flash ads.

We suggest using JavaScript to read the parameter from the document location URL and using it to set up the landing page for your clicks. You can use single quotes (') or double quotes (") in your code, but you can not mix them. Use either all single quote or all double quotes.

Backup/Default Clickthrough Parameter

To correctly call the clickthrough page through the backup ad image, you must supply a clickthrough parameter for that purpose. If you wish to also use that clickthrough URL in your HTML5 creative, you can configure the creative to use the Backup Clickthrough URL as the Default Clickthrough URL.

Retrieving the Click-Tag Parameter

The Click Tag URL can be retrieved using the following code:

```
function getParameterByName(name) {

name = name.replace(/[\[]/, "\\[").replace(/[\]]/, "\\]");

var regex = new RegExp("[\\?&]" + name + "=([^&#]*)"),
```

```
results = regex.exec(location.search);

return results === null ? "" :

decodeURIComponent(results[1].replace(/\+/g, " "));

}
```

Using Multiple Clickthrough URLs

By appending a URL escaped landing page to the click tag URL, our system will track the click and redirect the request to the given landing page.

```
var clickTag = getParameterByName(<Your Parameter Name>) +
encodeURIComponent(<Your Landing Page>);
```

Different landing pages can be used for different actions.

Using the Click-Tracking Parameter

When the advertisement is clicked on, the creative must direct the user to the correct click tag page in a new window.

Sample Creative HTML using ClickTAG

The following code block shows one way to retrieve and implement a creative with a clickthrough URL.

In this example, a single image is wrapped with an anchor tag which, when clicked, opens a new window with the address stored in the creative's `clickTAG` URL query parameter. The URL to open the window with is stored in `window.clickTAG` at line 11.

A creative may use any number of ways to open a new window to that address on click. This is one just one example.

```
<html>

<head>

<script type="text/javascript" charset="utf-8">

function getParameterByName(name) {

name = name.replace(/[\[]/, "\\[").replace(/[\]]/, "\\]");

var regex = new RegExp("[\\?&]" + name + "=([^&#]*)"),

results = regex.exec(location.search);

return results === null ? "" :

decodeURIComponent(results[1].replace(/\+/g, " "));

}

var clickTAG = getParameterByName("clickTAG");

//At this point, the value for the clickthrough URL is stored in the variable
"window.clickTAG"

//clickTAGs are case-sensitive, so make sure TAG/Tag/tag is uniform
throughout the creative and any supporting files

</script>
```

```
</head>

<body>

<a href="javascript:window.open(window.clickTAG);void(0);">

<img src="clickme.jpg"/>

</a>

</body>

</html>
```