ADITION

HTML5: technical specifications

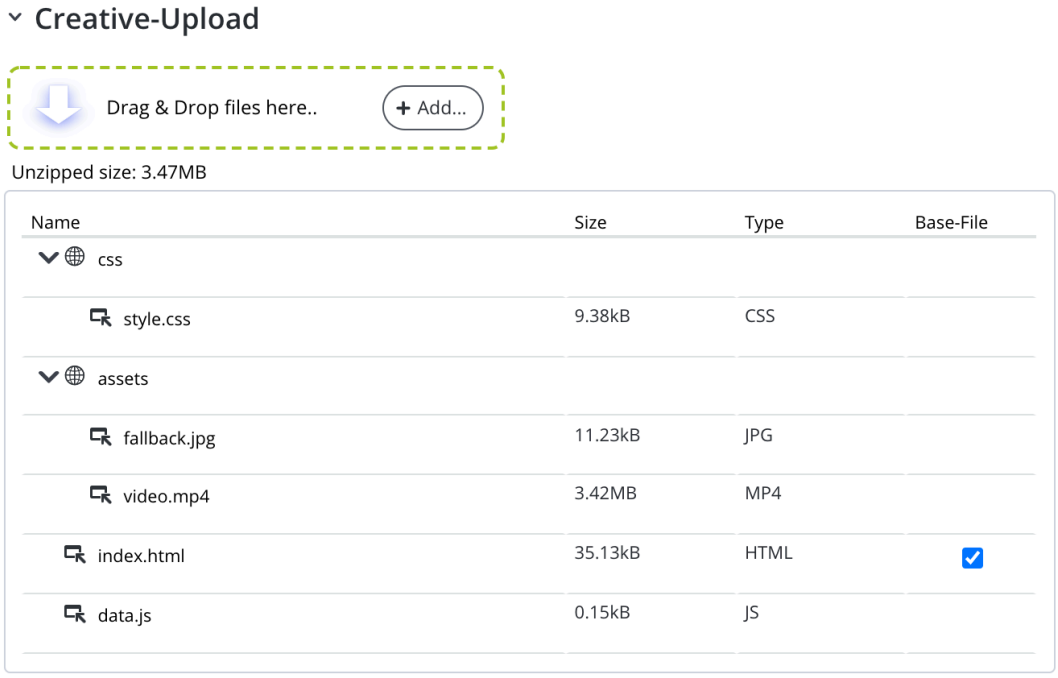
Version 2021-09-01

Content:

Introduction and limitations

To use HTML5 banners in ADITION, there are a number of requirements that must be taken into account.

HTML5 banners must be delivered in the form of a ZIP file (archive file type, file extension: .zip). These should contain all assets and files necessary to display the creative correctly and at least one HTML file (base file). The folder structure used within the ZIP is adopted by ADITION and can be viewed in ADITION after uploading the archive. The following is an example of this:



The following limitations apply:

Maximum file size of the ZIP:	10 MB
Maximum number of files and folders in the ZIP:	50
Allowed file types within the ZIP:	*.jpeg, *.gif, *.png, *.swf, *.flv, *.mp4, *.js, *.txt, *.html, *.css, *.xml, *.bin, *.webm, *.ogg, *.ogv, *.svg, *.ico, *.bmp, *.eot, *.ttf, *.woff, *.woff2, *.htm, *.jpg, *.json
Maximum number of clickable areas (clicktags):	10 (clicktag, clicktag1, clicktag2 [...] clicktag9)

clicktag implementation

ADITION expects so-called "clicktags" in the form of Javascript variables for clickable areas. If no clicktags are used or the URLs are fixed in the HTML5 banner, no click counting can take place via ADITION. See “Further notes on clicktags” for further troubleshooting. The implementation can be tested by clicking in the preview of the banner in ADITION. If clicktags have been used correctly, an intermediate page appears after the click with the following message:

Code examples for the use of clicktags

Simple hyperlink tag:

```
<html>

<head>
</head>

<body>
    <a href="#clicktag" id="clicktag" target="_blank">
        <div id="example_element" style="width: 300px; height: 250px">
        </div>
    </a>
    <script>
        document.getElementById('clicktag').setAttribute('href', clicktag);
    </script>
</body>

</html>
```

Important: the "example_element" area highlighted in blue represents the code of the actual banner or the elements of the banner. This part should not be copied if code is copied from the example above.

In this example, the banner is enclosed by an HTML anchor tag whose URL is set in a subsequent script via the clicktag variable.

For multiple clickareas:

```
<html>

<head>
</head>

<body>
    <a href="#clicktag" id="clicktag" target="_blank">
        <div id="example_element1" style="width: 300px; height: 250px">
        </div>
    </a>
    <a href="#clicktag1" id="clicktag1" target="_blank">
        <div id="example_element2" style="width: 300px; height: 250px">
        </div>
    </a>
    <script>
        document.getElementById('clicktag').setAttribute('href', clicktag);
        document.getElementById('clicktag1').setAttribute('href', clicktag1);
    </script>
</body>

</html>
```

Important: the "example_element1" and "example_element2" areas marked in blue represent areas of the banner that should be clickable. Those parts should not be copied if code is copied from the example above.

Further notes on clicktags

The clicktags can only be filled by the system if they are used in the base file (HTML file). If clicktags are used in external Javascript files, for example, they will not be filled with values by ADITION.

The clicktags cannot be recognized if they are written in a different notation, e.g. clickTag, Clicktag .

Therefore, please always use clicktags in lower case as shown in the example on page 3.

It is not necessary to declare Javascript variables. Therefore code snippets like the following should be avoided:

```
<script>
clicktag = "";
</script>
```

Furthermore, clicktags must not be filled with own values, like fallback URLs:

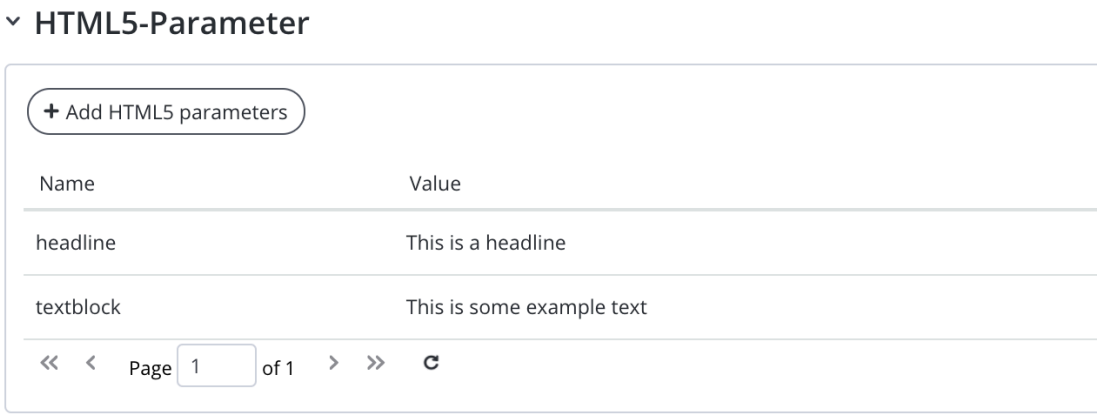
```
<script>
clicktag = "https://www.adition.com";
</script>
```

Both cases result in the values set by ADITION being overwritten and therefore no click counting will be possible.

HTML5 parameters

HTML5 parameters can be used to maintain information in ADITION that is processed in HTML5 banners. This means, for example, that text content can be specified and changed in ADITION without having to adapt the HTML files again.

The following is an example of the creation of HTML5 parameters in ADITION which will also be used as examples in the further course:



Processing HTML5 parameters in the HTML code

In the actual banner code (index.html of the HTML5 banner), the HTML5 parameters are available in the form of Javascript variables which are initially set by the adserver on delivery. The variable name is identical to the name of the HTML5 parameter in ADITION.

The following is an example of the use of these variables:

```
<p id="textblock_element"></p>

…

<p id="headline_element"></p>

…

<script>
document.getElementById('headline_element').innerHTML = headline;
document.getElementById('textblock_element').innerHTML = textblock;
</script>
```

Similar to what was noted before for the clicktags, the values of the variables must not be overwritten.

HTML5 Responsive Ads: Adserver adjustments

Banners with a dynamic size (responsive ads) require additional code in ADITION to ensure that the HTML5 iFrame of ADITION also receives a dynamic size.

The following code can be used for this purpose.

```
<script type="text/javascript">
(function() {
try {
var iframe_id = 'html5_iframe_%timestamp%';
var target = document.getElementById('ad_trgt_%timestamp%');
var bannerIframe = target.getElementsByTagName('iframe')[0];
bannerIframe.width='100%';
bannerIframe.height='100%';
bannerIframe.id=iframe_id;
bannerIframe.style.position='absolute';
}catch (exception) {
console.log('Failed to make banner in div id=ad_trgt_%timestamp% responsive. Err:' + exception);
}
})()
</script>
<script>var ad_vt_%timestamp% = {nodeId: "html5_iframe_%timestamp%"};</script>
```

The following points should be noted:

- This code must be stored in the field "Additional HTML code behind" in ADITION.
- The creative agency must develop the assets in a responsive manner / with dynamic properties, ADITION cannot provide sample code for this.
- The publisher must have adslots which provide corresponding sizes dynamically.
- Depending on the layout and setup of the publisher, it may be necessary to remove the following line:

  ```
  bannerIframe.style.position='absolute';
  ```

HTML5 Expandable Ads: Creative adjustments

For expanding banners it is necessary to communicate an expansion or contraction of the creative to the adserver, e.g. using postMessage.

For example, the following code could be used in the creative itself:

```
var adname = 'myAD';

function expandAd() {
    window.top.postMessage('expandAd:;:' + adname + ':;:' + expandedWidth + ':;:' +
expandedHeight, '*');
};

function collapseAd() {
    window.top.postMessage(
        'contractAd:;:' + adname + ':;:' + collapsedWidth + ':;:' + collapsedHeight, '*');
};
```

If this example is used, the following points should be noted:

- "myAD" can be replaced with your own meaningful name, but it is important here that the same name is not used by multiple banners at the same time on the target page. A random number/timestamp might help here.
- The methods "expandAd" and "contractAd" can be renamed and expanded upon, but the postMessage must be kept in its current form to cooperate with the following proposed adserver code.

In ADITION itself, the specified methods can then be caught and processed to customize elements such as the HTML5 iFrame.

HTML5 Expandable Ads: Adserver adjustments

If the previously suggested code is used on Creative side, the following code can be used on Adserver side:

```
<script>
    window.top.ovk = window.top.ovk || {
        ads2Handle: {}
    };
    var ovk = window.top.ovk;
    ovk.listenMessage = function(msg) {
        if (msg.data && msg.data.match(':;:')) {
            var call = msg.data.split(':;:');
            if (!ovk.ads2Handle[call[1]]) {
                ovk.walkFrames(call[1], window.top, msg);
            }
            ovk[call[0]](msg);
        }
    };

    ovk.walkFrames = function(adName, w, event) {
        var i, j, currentFrame;
        for (i = 0; i < w.frames.length; i++) {
            currentFrame = w.frames[i];
            if (event.source.window === currentFrame) {
                try {
                    for (j = 0; j < w.frames.length; j++) {
                        if (w.document.getElementsByTagName('iframe')[j].contentWindow ===
currentFrame) {
                            this.ads2Handle[adName] =
w.document.getElementsByTagName('iframe')[j];
                        }
                    }
                }
                catch (e) {throw e;}
            }
            if (currentFrame.frames.length > 0) {
                this.walkFrames(adName, currentFrame, event);
            }
        }
    };

    ovk.expandAd = function(msg) {
        var call = msg.data.split(':;:');
        this.origWidth = this.ads2Handle[call[1]].width;
        this.origHeight = this.ads2Handle[call[1]].height;
        this.ads2Handle[call[1]].style.width = call[2] + 'px';
        this.ads2Handle[call[1]].style.height = call[3] + 'px';
        if (this.ads2Handle[call[1]].contentWindow.parent !== window.top) {
            try {
                this.ads2Handle[call[1]].contentWindow.frameElement.style.width = call[2] +
'px';
                this.ads2Handle[call[1]].contentWindow.frameElement.style.height = call[3] +
'px';
            }
            catch (e) {}
        }
    };

    ovk.contractAd = function(msg) {
        var call = msg.data.split(':;:');
        this.ads2Handle[call[1]].style.width = this.origWidth + 'px';
        this.ads2Handle[call[1]].style.height = this.origHeight + 'px';
    };

    (window.attachEvent) ? window.attachEvent('onmessage', ovk.listenMessage) :
window.addEventListener('message',
        ovk.listenMessage, false);
</script>
```

If this example is used, the following points should be noted:

- This code must be stored in ADITION in the field "Additional HTML code before".
- This only changes the size of the adserver elements of ADITION during expansion or contraction. Depending on how the ad is rendered on the publisher side and which reference point it utilizes in the layout, it expands to the left or right. ADITION has no influence on this.