

Ad delivery:
display-campaign@ad-alliance.de

last updated: 13st February 2026

Enquiries:
displayspecs@adalliance.atlassian.net

HTML5 Guidelines

Content

Ad formats – HTML5 ad formats	2
File size - redirect or zip-archive incl. code and companion files	2
HTTPS	2
File number and code compression	2
File hierarchy and folder structure	2
Shared libraries	2
Back-up image, browser compatibility	3
Click tag	3
Grafics compression	4
Video	4
Animation	4
Sound	4
Frameworks, libraries and namespaces	4
Delivery	5
Pre-run time	5
document.write	5
Special considerations for mobile ads	5
Other sources of information	5

Ad formats – HTML5 ad formats

Mobile or desktop ads in the Ad Alliance network use the Google Library and will be delivered in “friendly iframes”.

File size - redirect or zip-archive incl. code and companion files

In order to ensure that the set-up of the media website and the ad is not delayed unnecessarily, creative staff are required to keep individual elements of the HTML5 ad as small as possible with respect to number and file size and to adhere to the following mandatory limits.

The full file size embraces all elements that are being transferred to the client („zip files“).

The elements are loaded in **three different stages**: initial load, host-initiated subload und user-initiated load.

Initial load

- **desktop / tablet – 200 kB**
- **mobile – 200 kB**
- **max. amount of requests: 15**

This loading type describes the sum of all zipped assets that are needed for an initial ad display containing graphics, HTML-, css-, java script files and all shared libraries or customized files.

Host-initiated subload (for Rich Media)

- **desktop / tablet – 300 kB**
- **mobile – 200 kB**

This loading type describes the sum of all zipped assets that are being loaded after the load event and before user interaction.

The creative must wait for this event to happen and should use event listeners before execution.

User-initiated load (polite load)

- **desktop / tablet – unlimited* (except for Video)**
- **mobile – unlimited* (except for Video)**

This loading type allows unlimited (video is, however, limited) zip-file loading after user interaction.

***Therefore user experience and creative loading as well as creative environment must always be payed attention to.**

HTTPS

Please note that all components of a creative (i.e. script, iframe, redirect tags, third party tracking pixel and other externally hosted resources) are to be delivered as HTTPS-compatible components. This is to ensure our ad quality standards and to prevent display or measurement errors as well as to avoid violations of security settings across different browsers.

File number and code compression

The number of files/elements for an HTML5 ad must be kept as low as possible in order to minimize server processes/requests. Accordingly, code compilation methods and code summaries must be included in a single file.

File hierarchy and folder structure

The number of levels/folder structure must be kept to a minimum, max. one level (i.e. images, scripts).

Please use only letters and numbers when naming files and folders, as we cannot guarantee the required functionality otherwise.

The file and folder names must not contain special characters or spaces.

Shared libraries

We recommend the use of shared libraries to enjoy the benefit of the cache function in each browser, so that already loaded libraries must not be charged a second time. Thereby, the charging time of ads and the site can be reduced. See <https://developers.google.com/speed/libraries/>

Back-up image, browser compatibility

If a browser does not support a special feature or library that is used in the creative, a fallback JPG/GIF graphic (file size according to initial load file size) should be displayed. For example, if the ad is not supported by Edge, the agency must ensure that the fallback is displayed in this browser.

The ads must be tested by the creative agency on all standard browsers and the marketer (Ad Alliance) must be informed of any browsers that need to be excluded. In addition to that we shall have the right to exclude browser versions from designated ads.

Click tag

The click tag must be written as follows: **clicktag**.

Please point out to us if you have chosen a different name and / or use multiple click areas. The following lines of code must be integrated into the index.html of the HTML5 creative in the sequence given below so that our clicktag is being passed through:

```
<script>
var getUriParams = function() {
    var query_string = {}
    var query = window.location.search.substring(1);
    var parmsArray = query.split('&');
    var clickURL = null;
    if(parmsArray.length <= 0) return query_string;
    for(var i = 0; i < parmsArray.length; i++) {
        var pair = parmsArray[i].split('=');
        var val = decodeURIComponent(pair[1]);
        if(i===0){
            clickURL =val.split('adurl=')[0] + 'adurl=';
        } else if(i > 0 && clickURL !== null && val.indexOf('adurl=') === 0) {
            val = clickURL + encodeURIComponent(val);
        }
        if (val != " && pair[0] != ") query_string[pair[0]] = val;
    }
    return query_string;
}();
</script>
```

This function returns all GET parameters that are passed to the file.

Example of HTML and assignment of links at the end of the HTML document:

Designation of the links (without hard-coded URLs) – example of **2 HTML links without assigned links**:

```
<a href="#clicktag" id="IAB_clicktag" target="_blank">IAB clicktag</a>
<a href="#clicktag2" id="IAB_clicktag2" target="_blank">IAB clicktag</a>
```

Afterwards these two JavaScript lines allow the click tags to be assigned to the HTML elements:

```
<script>
    document.getElementById('IAB_clicktag').setAttribute('href', getUriParams.clicktag);
    document.getElementById('IAB_clicktag2').setAttribute('href', getUriParams.clicktag2);
</script>
```

The HTML5 ads should be tested by the creative agency to ensure the functionality of click tag passes, in order to prevent unnecessary feedback loops. Therefore the index.html must be called with a browser in the following form: ***index.html?clicktag=%LANDINGPAGE%***.

%LANDINGPAGE% should be replaced with a test destination page and must be passed with URL encoding (encodeURIComponent-Function).

With proper implementation the ad should now be clickable and passes through to the landing page.

Grafics compression

The file size of graphics must be optimized. We recommend using PNG Crusher and scalable vector graphics.

Video

Videos in HTML5 ads are integrated using the tag `<video></video>`.

Ads with videos must be provided with a preview image. The video starts as soon as it has loaded on desktop devices or following user interaction on mobile devices.

Please note that click tags cannot be placed on videos on mobile devices. The click tag must be placed on an area outside the video.

The quality and file size of the video must be optimized, and it must be streamed in a suitable server environment.

max. video file size at autostart (without sound) = 4 MB (polite load)

max. video file size when activated by the user ("on click") = 15 MB (polite load userinitiated)

The video must be provided as mp4 in 16:9 format.

The stream may loop a maximum of 1x.

The video length must not exceed 30 seconds.

Example of code:

```
<video controls height='640' width='360'>
  <source src='yourVideo.mp4' type='video/mp4' />
  <source src='yourVideo.webm' type='video/webm' />
</video>
```

Animation

Please ensure that animations do not place an unnecessary load on the client CPU. Autostart is only allowed without sound.

Avoid using multiple parallel animations and overlapping transparent graphics. CPU and GPU load should be taken into account when selecting CSS3 or JavaScript animations.

Sound

Start of sound is only allowed by user interaction ("on click"). Autostart is forbidden.

The user must always have the opportunity to switch on/off the sound independently from the length. Sound may loop only 1x. Regarding the volume, the loudness standard EBU 128 must be observed.

Frameworks, libraries and namespaces

When integrating frameworks and libraries (e.g. bootstrap), ensure that they do not produce conflicts with the website functions.

When using global JavaScript variables, these should have their own namespace or else be clearly designated, e.g. “**adaAdtext**” instead of just “**text**”. This clear designation requirement also applies when assigning HTML IDs (e.g. **adaExpandableWrapper** instead of just **wrapper** for <div> containers) and ID, class and type selectors in CSS (e.g. **.adaClickAreas** instead of just **.clicks** in CSS). Otherwise, site-specific variables/designators/CSS styles may negatively impact on the display and function of the ad and vice versa.

Delivery

There are several options to deliver HTML5 ads. In all cases standards for file size and number of files apply.

- Delivery of a basic HTML / script, all assets are hosted by the agency/customer, there are only absolute URL's used
- Delivery of redirects
- Delivery of a basic HTML / script, all assets are hosted by Ad Alliance, there are only relative URLs used. This does not apply with expanding or overlapping creatives like:
 - layer, mobile interstitial and mobile pre-expand and only when these ads use the Ad Alliance-own container to expand and close the creative
- In case that Ad Alliance is taking over the hosting, all elements that are required must be in a compressed zip archive except externally hosted libraries and fonts. The zip archive must contain an index.html file (main file) and all embedded scripts and elements must be relatively linked. The file and folder names must not contain special characters or spaces.

Pre-run time

Ads must be delivered with a pre-runtime of three to five working days (see pre-run time in individual specifications). This deadline must be taken into account during campaign planning.

document.write

The use of document.write has to be avoided because ads are loaded and executed in asynchronous environments. Alternatively DOM operations like appendChild () can be used, executed by a document.onload type handler.

Special considerations for mobile ads

- All levels of the ad must have an ad identifier (-w-), i.e. both the mobile content ad and any expand/layer areas.
- For ads that wrap themselves above the content in full screen format without prior user action (preexpandable), a distance of 100px must be adhered to the top of the screen to make the header of the site / app visible.
- All JavaScript-based ad formats must have a no-script area that ensures a graphic fallback ad.
- When using jQuery, the ad must check for any framework that has been already loaded by the site (and in case of doubt, it must use this framework). Your own framework can only be used if the site does not use jQuery.
- The ad must independently identify the orientation of the device and if required, independently adjust its visuals and functions to any shift between portrait and landscape.
- Unless otherwise agreed, hosting and tracking must be carried out externally.
- **MRAID compliance must be ensured for in-app delivery.**
<http://www.iab.com/guidelines/mobile-rich-media-ad-interface-definitions-mraid/>

Other sources of information

Information from the IAB:

<https://www.iab.com/guidelines/html5-for-digital-advertising-guidance-for-ad-designers-creative-technologists/>
<https://www.iab.com/wp-content/uploads/2016/04/HTML5forDigitalAdvertising2.0.pdf>

Ad Alliance specifications:

<https://www.ad-alliance.de/portfolio/in-page/display/spezifikationen#c3194>

More around HTML5:

<http://caniuse.com>