

Ad Tags

Ad Tags



Unknown macro: 'show-if'

Ad tags are used for many purposes in display advertising:

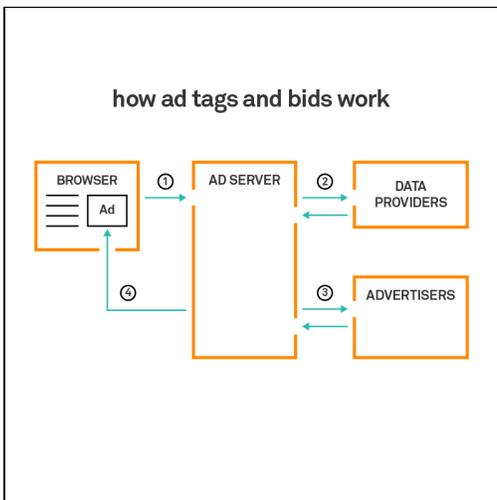
- Publishers place ad tags on a website in order to sell ad space
- Advertiser use ad tags to direct the browser to a particular image or flash creative
- Ad servers use ad tags to move ad calls along to the next step in the decisioning chain
- Third-party data providers use ad tags to collect segment information about users

These tags are all essentially the same in structure: they are all requests for content of a certain size and type from a certain URL. The content is either a creative or another ad tag and it may be returned immediately in one step or after multiple steps (an auction, redirects, etc.) each with its own tag. An ad tag may look very long and complicated if information about the ad call is included in the tag itself, or it might be very short and simple if ad call information is stored in the server to which the tag's URL is pointed.

On This Page

- [How Ad Tags Work](#)
- [What Does an Ad Tag Look Like?](#)
- [Further Examples](#)
- [Related Topics](#)

How Ad Tags Work



Here's a simple example of how ad tags move through different systems. The example uses one ad server but in reality publishers and advertisers may each have their own ad servers: the publisher to track the impressions and the advertiser to determine which ads to serve.

1. When the user visits a publisher web site, the browser sends an ad tag to the ad server. This ad tag contains information about the user and the ad placement.
2. The ad server optionally passes an ad tag to a third-party data provider to retrieve information about user segmenting or contextual targeting.
3. The ad server then passes an ad tag to the advertisers. Depending on the publisher-advertiser relationship, this may mean simply directly requesting ads for guaranteed buys, or it may involve requesting bids from multiple advertisers and carrying out an auction to determine the most profitable result for the publisher.
4. The ad server delivers the creative (the ad) to the browser. Typically, this means returning an ad tag with a creative URL, with the creative itself hosted on an independent content server. For more information about creatives, see [Creatives: An Overview](#).

What Does an Ad Tag Look Like?

An ad tag has two parts:

- A URL, from which the browser will request content
- Some HTML and possibly some JavaScript code. (HTML lets you create static webpage content. JavaScript is designed for performing

dynamic tasks.)

The purpose of the code is to tell the browser how to display the ad (or other content) that they get from the URL request. For example, the HTML `<iframe>` tag tells the browser to open a mini browser window of a specified size inside the current window. This way the ad content cannot expand beyond the size specified and "take over" the screen.

For example, here's a Xandr ad tag that a publisher would use to auction an impression.

```
<script src="http://ib.adnxs.com/ttj?id=1234" type="text/javascript"></script>
```

This is what the different parts of the tag are doing:

- The HTML `<script>` `</script>` and `type="text/javascript"` tell the browser that JavaScript code will be executed here. The browser needs to be alerted to this so it can process the JavaScript correctly, rather than treating it as HTML.
- The HTML `src=` tells the browser to retrieve content from the URL `http://ib.adnxs.com/ttj?id=1234`
- The URL points to the Xandr impression bus, which processes all requests to the Xandr platform. `/ttj` is a Xandr designation for a JavaScript call, and `id=1234` is the ID that Xandr has assigned to this ad tag. This allows information about the ad tag, such as ad size or reserve price for the ad, to be stored on the impression bus rather than on the page itself. This way the information can be changed at any time without requiring a new tag.

When the Xandr impression bus receives the tag, it runs an auction. Something like the following raw JavaScript code is returned to the browser:

```
document.write('<iframe frameborder="0" width="160" height="600" marginheight="0" marginwidth="0" target="_blank" scrolling="no" src="http://ad.yieldmanager.com/st?ad_type=iframe&ad_size=160x600&section=560122&m6li=1302146"></iframe>');
```

This is what the different elements of the JavaScript are doing:

- `iframe frameborder="0" width="160" height="600"` tells the browser to open a 160x600 iframe.
- `src="http://ad.yieldmanager.com/st?ad_type=iframe&ad_size=160x600§ion=560122&m6li=1302146"` tells the browser to deposit specific content from the Yield Manager ad server into the iframe. The Yield Manager URL does not point to an actual creative image file, because the Yield Manager ad server makes a dynamic decision about which image to pass to the browser. This URL's query string (everything after the "?") gives Yield Manager information that will help it decide which creative to pass back.

Most people dealing with ad tags (publishers, who need tags to put on their inventory pages, and advertisers, who may use tags to direct a browser to their creative) aren't hand building tags; they are inputting their page information or creatives into their ad server's user interface, which creates a properly formatted ad tag for them.

Further Examples

There are many permutations of ad tag syntax depending on what the tag is doing and whether the ad tag info is on the page or in a server.

>> A publisher ensures that the ad is an iframe.

```
<iframe src="http://ib.adnxs.com/tt?id=1234" width=160 height=600 scrolling=no frameborder=0 padding=0></iframe>
```

>> A Google AdSense tag

```
<script type="text/javascript"><!--
google_ad_client = "pub-11111111";
/* 160x600 - TOP */
google_ad_slot = "12345";
google_ad_width = 160;
google_ad_height = 600;
//-->
</script>
<script type="text/javascript"
src="http://pagead2.googlesyndication.com/pagead/show_ads.js">
</script>
```

Even though it looks like a piece of the tag is commented out with "!--", this isn't really the case. Some scripting engines, including those for JavaScript, allow the script statements to be enclosed in a comment. Then a browser that doesn't recognize the JavaScript element will ignore the comment, but others will execute it. This particular faux comment is passing information to the Google ad server in order to determine the right ad to show on the page.

>> A publisher provides alternate content for browsers with JavaScript disabled.

```
<SCRIPT type="text/javascript"
SRC="http://ad.advertiser.net/adID?1234;sz=184x90;click=http://www.publisher.com/page.
html">
</SCRIPT>
<NOSCRIPT>
<A HREF="http://www.publisher.com/page.html">
<IMG SRC="http://ad.advertiser.net"></A>
</NOSCRIPT>
```

In the NOSCRIPT section, the HREF link is the landing page of the ad and the SRC is the ad itself.

Related Topics

- [Creatives](#)