

Choosing the right tag

By [Saskia Delfgaauw](#), issued by [Incubeta](#)

8 Nov 2012

Why to implement the new Google Publisher Tag, and how to choose the right GPT modes for your website.

Using the Google Publisher Tag will have many benefits for publishers, for example serving rich media ads will become easier. One of the primary decisions you need to make when tagging your pages with the Google Publisher Tag (GPT) is whether to use asynchronous or synchronous rendering. You'll also need to decide whether you want to fetch all ads in a single request and which ad units and targeting criteria to apply. Generally, we recommend using asynchronous rendering with single-request mode enabled. This combination offers the best page load experience and guaranteed roadblocks. But let's start with explaining what the new GPT is and why it is better than the DART tag.

What is the Google Publisher Tag?

The Google Publisher Tag (GPT) is an ad tagging format that provides you with a number of benefits and improvements over the DART tag. The greater length of the tag is not a reflection of its overall load time; in fact, you can expect faster performance and load time from Google Publisher Tag than from DART tags. The DFP upgrade provides your organisation with a scalable inventory structure that allows you to manage, target, and report on campaigns across your existing inventory channels, without needing to re-tag your sites. Your DART tags will continue to work without any changes.

How does the Google Publisher Tag work?

The Google Publisher Tag is used to define available ad slots on your website. Placing a tag on a page creates a communication path between the ad server and a user's browser. When a page containing a GPT is rendered, the following sequence of events occurs:

1. A request is made from the user's browser to the ad server for gpt.js, the tagging JavaScript
2. The tagging JavaScript builds and sends one or more requests (depending on whether single-request mode is enabled) to the ad server for ads tagged on the page
3. The ad server recognizes the ad units and any custom targeting contained within the request
4. The ad server selects and returns the best matching ad
5. The JavaScript code associated with the ad tag displays the ad on the page

Does the Google Publisher Tag work with https:// secure pages?

Yes. The Google Publisher Tag works automatically with secure web pages whose URLs begin with <https://>. There's no need to modify the tag in any way for serving on a secure page.

What's asynchronous rendering, and why should you use it?

An asynchronous fetch means that the GPT code on your page does not block the HTML (content of your webpage) when other data such as display banners are still loading. For example, if you have a rich content leaderboard that takes time to load, with asynchronous tags the rest of the page will load while the creative is loading. Your visitor will see the content of the webpage while the rich media ad is still loading. This will result in a better user experience.

You should use asynchronous rendering if you want to implement video companion dynamic allocation. This does not work with synchronous rendering.

There are two levels of asynchronous loading with GPT:

- Asynchronous loading of the GPT JavaScript library. Using asynchronous mode (called in the tag) means that the section of your page will not wait for the GPT JavaScript library to load before rendering.
- Asynchronous rendering of the creatives in the section of the document. This allows HTML elements to load without waiting for the creatives before them to render.

We recommend that you load both the library and the creatives asynchronously in order to get the greatest performance benefit and user experience. However, it is possible to load the library synchronously but render the creatives asynchronously.

Can I use asynchronous tags on some pages and synchronous on others? Can I use both tags on the same page?

You can use asynchronous tags on some pages of your website and synchronous tags on others. You want this if you typically use asynchronous tags but are running a campaign that uses rich content ads that do not work well with friendly iframes. In that case, you could use synchronous tags only on the pages to which that particular rich content ads campaign will serve, and asynchronous tags everywhere else. You cannot mix asynchronous and synchronous tags on the same page. Serving 3rd party out of page material in GTP Asynchronous Tags is possible. GPT asynchronous uses iframes to serve ads. The iframe is necessary for asynchronous operation. The iframe preferably is a friendly iframe.

Create (ask for) iframe friendly rich media and prevent whitespace When you serve ads in asynchronous GPT's you should use friendly iframes. Why? GPT uses friendly iframes to serve ads, which is necessary for asynchronous operation. Some ads may not display correctly in frames. Examples include expandable ads that expect to be written directly on the main page or creatives that attempt to access the DOM elements (and the JavaScript environment) directly on the page. Additionally, if you are using a third-party creative with a size that differs from your ad slot, the iframe may cut off the ad or cause extra whitespace. There are a few things you can do to ensure that your ads display properly in GPT's asynchronous mode:

- [Convert custom templates](#) to work with friendly iframes and work with your rich content providers to establish or obtain proper iframe-ready ad tags
- [Read the IAB's list of best practices](#) for helping create iframe-friendly rich content ads. If these best practices are followed, most rich content ads should render properly, even in asynchronous mode

Implement one of two common methods for enabling expanding and floating creatives to work properly with iframes: friendly iframes (recommended) and iframe busting files, both of which are described below.

What to do when vendors don't support friendly iframes? Bust the iframe

When the rich content vendor (e.g. Eyeblander, Pointroll, DoubleClick Studio) doesn't support friendly iframes they should provide you with an HTML iframe busting file. This HTML file needs to be placed on your server, usually the same one that serves your web site. In general, once the file is on the server, it can be used for all ads from that rich content vendor. With this implementation, the creative is on the main page and is able to access the page content. This is a security or privacy concern for some publishers. When the ad serves inside of an iframe, it creates an additional iframe. The additional iframe calls the iframe busting file. The ad then uses the iframe busting file to place the creative on the main page. The end result is the creative is on the page, outside of the iframe used to serve the ad code. Most rich content vendors support this workaround; ask yours if they do!

iframe busting file advantages:

- Supported by most rich media vendors
- Supported by GPT asynchronous disadvantages
- Possible security or privacy concerns

- Publisher needs to host an iframe busting file for each rich media vendor.

To implement this method, you, the publisher, need to host the iframe busting file on your site's server. A separate file is used for each rich media vendor. In general, once the file is on the server, it can be used for all ads from that rich media vendor. How to use a friendly iframe A friendly iframe is on the same domain as the main page. So any ad served into a friendly iframe can access the page content, even if the ad does not escape the iframe. For some publishers, using friendly iframes is a security or privacy concern. To implement this method, the publisher must serve ads with friendly iframes. Switching to friendly iframes might involve redesigning how ads are served on the site. The rich media vendor must also be able to support escaping friendly iframes without an iframe busting file. friendly iframes advantages - Iframe busting file not needed - IAB has published recommendations for friendly iframes - Supported by GPT async disadvantages - Possible security or privacy concerns - Supported by a limited number of rich media vendors. Some sites use iframes that are on the same domain as the main page. For example, the site is at <http://www.google.com> and the iframe containing the ads is at <http://www.google.com/ads>. When the main page and the iframe are on the same domain, the iframe is a friendly iframe.

Some rich media vendors (DoubleClick Studio) can escape friendly iframes directly, without an iframe busting file. Inside of the iframe, a Javascript tag is used to serve the ad. So the ad serves inside of the iframe. In order for the entire creative to be visible, it still needs to escape the iframe. The ad code sees that a friendly iframe is used and then places the creative on the main page. This method is similar to the iframe busting file method. The main difference is that this method skips the step with the additional iframe that calls the iframe busting file. The end result is the creative is on the page, outside of the iframe used to serve the ad code, and an iframe busting file is not needed. Only some rich media vendors support escaping friendly iframes without an iframe busting file. If the vendor does not support this variation, it is still possible to use an iframe busting file with friendly iframes. The rich media vendor needs to provide the iframe busting file. Choose the single request mode to guarantee roadblocks.

What's single-request mode and why is it recommended? Single-request mode tags call all ads, for example several remarketing scripts, at once in the header or footer, rather than requesting each ad separately inline with the ad slot. We recommend using single-request mode because grouping all ad calls into a single request allows you to guarantee roadblocks (serving all creatives from a line item together on the same page). Additionally, your page load performance may benefit from a reduced number of requests.

When not to choose the single request mode? Cases where not using single-request mode may be preferable when all DFP creative types and line item types are supported by single-request mode. However, there are some limited cases that are not supported: DoubleClick tag rewind: A DoubleClick tag causes the ad server to retrieve a different ad from inventory that belongs to a DFA or DFP network.

With rewind, if an ad on the target network is not available, an ad will be retrieved from the DFP network making the ad call instead. This rewind feature does not yet work in single request mode. Google programmable ads are not compatible with single-request mode. Can I use different tagging styles on the same network? On the same page? It's OK to use different tagging styles across your site. However, we don't recommend using different tagging styles on the same page. The DFP ad server may not be able to correlate multiple request types and could end up serving duplicate ads.

What are the benefits of using Google Publisher Tags?

- Five-level inventory hierarchy: The Google Publisher Tag allows you to use more granular levels of inventory in the DFP front-end (DART tags allow only two levels). With five levels of hierarchy, you'll be able to create much more specific targeting based on your site content. For example, you could create inventory on an electronics site to target the following structure:
- Electronics > TVs > LCD > Brand > Under_1000
- Faster page loads: An asynchronous JavaScript fetch means that instead of waiting for the JavaScript to be returned from the DoubleClick servers, the page continues rendering and loads the ads into iframes when the creatives are returned from the server. Google Publisher Console: The tag comes with a built-in debugging and support tool called

the

- Google Publisher Console, which is enabled on all pages containing the Google Publisher Tag. To activate the Publisher Console, load your webpage containing Google Publisher Tags into a browser and append ? google_console=1 to the URL and use the keyboard shortcut Ctrl+F10 to toggle the console. The console provides checks for common tagging errors, visual highlights of all ad units and creatives on the page to help with debugging, and an alternative point of entry into the DFP front-end.
- Single request mode: Instead of sending individual ad requests to DoubleClick servers, the browser is able to send one request notifying the server of all ad units on the page. This enables advanced roadblocking and improves page load time.
- Automated setup for interstitials: DFP lets you specify that your tags are for an out-of-page unit and automatically adds the additional code. There's no need to add code manually.

For more information please contact Andries de Jonge andries@dqna.com

<http://www.dqna.com/en/blog/99-choosing-the-right-tag.html>

" **4 reasons why CFOs love data-drive creative** 26 Apr 2024

" **Why data-driven creative projects fail and 3 steps to ensure success** 17 Apr 2024

" **Jaco Lintvelt appointed as the new managing director of Incubeta Africa** 11 Apr 2024

" **Data driven creative - when marketing is indistinguishable from magic** 13 Nov 2023

" **Victoria Webb appointed as new managing director of Incubeta MENA** 24 Oct 2023

Incubeta



Incubeta is an international team of experts in marketing, technology, data, and creative.

[Profile](#) | [News](#) | [Contact](#) | [Twitter](#) | [RSS Feed](#)

For more, visit: <https://www.bizcommunity.com>