

Tracking AdWords Conversion On Click Event

Written by dcPages

If you have ever gotten really serious about a Google AdWords account, then you have definitely spent some time setting up AdWords Conversion Tracking. Conversion tracking is so essential to effectively managing an AdWords account that I won't run one without it. In the standard case, it is very easy to setup, you create a conversion, paste the code into your page and you're done. Unfortunately things don't always work out that easily. What if your conversion isn't defined by viewing a page, what if it's downloading a PDF, watching a video, clicking an affiliate link, etc? That's where things get a little bit trickier. I almost lost a client due to my demand of conversion tracking, and that is what forced me to figure out how to track conversions without displaying a "thank you" page.

This article will take you through the process that I use to track conversions using onclick events. Note: the method I prefer relies on jQuery. (Let's face it jQuery really does let you write less and do more). If you would like to learn how to track AdWords Conversions on click events WITHOUT jQuery, then skip to the end of this article to see that snippet of code.

To start out with, let's take a look at the default AdWords Conversion Tracking Code:

```
<!-- Google Code for Contact Page Conversion Page --> <script type="text/javascript"> /*
<![CDATA[ */ var google_conversion_id = XXXXXXXXXXXXX; var google_conversion_language
= "en"; var google_conversion_format = "3"; var google_conversion_color = "ffffff"; var
google_conversion_label = "XXXXXXXXXXXX"; var google_conversion_value = 0; /* ]]> */
</script> <script type="text/javascript"
src="http://www.googleadservices.com/pagead/conversion.js"> </script> <noscript> <div
style="display:inline;">  </div> </noscript>
```

The code here is pretty straight-forward, you set some variables so that Google knows that type of conversion pixel to display, things like ID, colors, labels, etc. Then you call a script from one of Google's domains which takes the variables you just set and displays a conversion pixel (and the little privacy notification if you selected that option when you created the conversion). The trick here is that you can just call a function when someone clicks on your download link to track that download as a conversion in AdWords, because you have to actually call that "conversion.js" from Google's server in order to get it to track.

//

The answer to tracking conversions on click events turns out to be a little counter-intuitive. Clearly, we'll need to use some sort of Javascript to get this done, but our Javascript code will actually use the <noscript> portion of the AdWords Conversion script. Most people never look at

Tracking AdWords Conversion On Click Event

Written by dcPages

the <noscript> part. I even know a developer who leaves it out when he sets up conversion tracking, because his business depends so much on technology that if you don't have Javascript enabled, then you're not his target market (even if you do sign up for his newsletter), but I digress...

If you take a look at the <noscript> part of the AdWords Conversion Script, you'll see that all it does is display an image, with a source URL that uses the same 'google_conversion_id' and 'google_conversion_label' from the traditional script. With that in mind, we can track conversions just by displaying an with the right URL in the source. Pretty simple, right? OK, so on to the code.

To display an image with Javascript is pretty easy, and I've seen several snippets out there that all do basically the same thing. Create an image, assign it a source and you're all set:

```
var image = new Image(1,1); image.src =  
"http://www.googleadservices.com/pagead/conversion/XXXXXXXXX/?label=XXXXXXXXX&am;gu  
id=ON&am;script=0";
```

The issue here is that I've experienced some inconsistencies in performance with that script. In addition, if you like to use developer tools (like "Inspect Element" in Chrome or Safari) then the method above doesn't make it very easy to see if the pixel is actually getting displayed.

In order to fix the issues that I had run into with the above method, I figured I needed to specify exactly where on the page I wanted the conversion pixel so that I could easily check that it is working using my browser. The code to display the pixel on a click is very simple, especially since we're using jQuery, so I'll just paste it in here and then we can go through it in more detail for those of you who want to know what's really going on here.

```
$(document).ready( function() { $("#emailLink").click( function() {  
$(this).parent().append('<div style="display:inline;"></div>'); _gaq.push(['_trackPageview','/email']); });
});
```

That's all it is, just 6 lines, and only 2 of those lines of code actually do any real work for us. The first line is the standard jQuery wrapper, which waits until the page is done loading and the DOM is ready before creating the click function. The second line declares a function that will fire when the user clicks on an element with the id "emailLink" (yes, the script I copied this from is setup to track when people click on a mailto: link). Now the really fun stuff starts. The third line of code grabs the parent element of our "emailLink" and appends the code from the <noscript> portion of the Google AdWords Tracking Code.

```
//
```

Technically, you could stop there. Now that you've output the AdWords Conversion Pixel, you're job is done, but I also added a fourth line of code to also track a pageview in Google Analytics (so that I can track this as an Analytics goal as well). Note: you could fire the Google Analytics part as an "Event" rather than a pageview, like this:

```
_gaq.push(['_trackEvent','Email Link', 'Email Link Click']);
```

I often choose to track things like a PDF as a pageview, since the user is actually viewing something (often in their browser if they have an Adobe plugin), and track button clicks as events.

So that's it, just a few lines of code and you've got it. The trick was to create some custom Javascript to display the <noscript> portion of the Google AdWords Conversion Tracking Script. Now that you know the secret, the sky is the limit. I've personally used this for many different actions, including:

- PDF Download
- Mailto: Link
- Affiliate Link
- Playing a Video
- Linking to a 3rd-Party Shopping Cart
- Ajax Forms

I think that last one has become the most important implementation of this script. Some clients

Tracking AdWords Conversion On Click Event

Written by dcPages

have the need for atypical conversions, but almost all of them want to track a form submission of some kind. In the old days, that was easy, just put the AdWords Conversion Tracking Script on the "Thank You" page that the user saw after they submitted the form. These days, more and more clients want fancy, web 2.0, on-page interactions. That means forms that post with Ajax and return a result without re-loading the page. With the above script, that doesn't pose any particular problem for AdWords Conversion Tracking. All you need to do is display that pixel when the form is successfully submitted via Ajax.