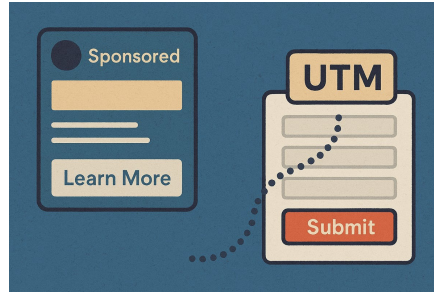


ALLAN THRAEN | 1 days ago | PDF |

Optimizely (Episerver) CMS Website Improvements Tips and Tricks C#

TRACKING UTM PARAMETERS IN OPTIMIZEZY CMS FORMS



When you are spending your marketing dollars on social media / CPC campaigns, correctly attributing your leads is everything so you know where to invest more. Usually you can get this insight from your Marketing Automation or analytics - but I recently got a question if it's possible to also automatically add it to your Optimizely CMS forms. And of course it is. Here are two ways of doing that.



Want to get more out of Optimizely CMS (Episerver) Forms?

We are experts in optimizing and extending Optimizely CMS forms. Reach out and let's talk.

REACH OUT

NOTE: This blog post is written for CMS 12. However with minor tweaks it can also be used in CMS 11 and earlier.

If you're running paid campaigns or just want better insight into where your leads are coming from, tracking UTM parameters is a must. UTM parameters – like `utm_source`, `utm_campaign`, and `utm_medium` – help you attribute conversions to the right traffic sources.

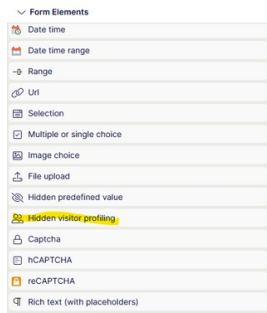
In Optimizely (formerly Episerver), tracking UTM data in Forms isn't built-in, but there are a couple of ways to get it working. In this post, I'll show you two simple solutions:

1. A **no-code** method using Audiences (previously Visitor Groups)
2. A **code-based** method using a datasource that can extend your forms with a hidden field that will include the UTM parameters in the submission.

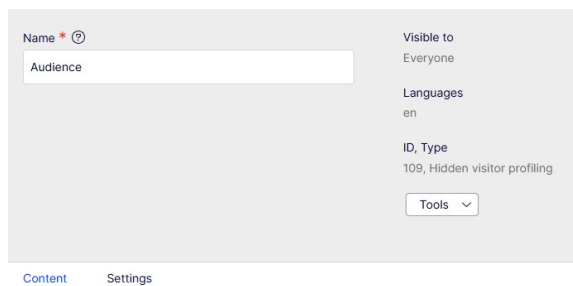
Let's dive in.

Quick and easy - no-code solution

While forms doesn't come with built-in support for adding query-line parameters to your submission, it does come out of the box with a neat little formelement called "Hidden Visitor Profiling".



This little hidden gem let's your add hidden fields to your form with dynamically collected visitor information. We'll essentially use this in both approaches - but out-of-the-box this also comes with support for including a list of audiences (visitor groups) that the visitor is a part of. So, essentially, by just adding this we can get a comma-separated list in our form submission with all the audience groups.



Auto-collect visitor data ⓘ

Geo
 Profile
 User agent
 Audiences

Name

Now, all we need to do is basically to make sure that visitors that land on your site with certain UTM parameters will be in a corresponding and appropriately named audience. We can do this by using the "Landing Page" criterion which is also built-in.

Edit Audience
 Adapt content on your website by first creating audiences and then using the audiences to target the content on pages.

Name *

Notes

Enable statistics for this audience
 Make this audience available when setting access rights for pages and files

Match

Landing URL ⓘ

Now, with those two additions, an extra field will be added to all form submissions with the value "UTMSource:Facebook" on those submissions that landed on your site following a link on facebook. And of course this could be used for any UTM parameters (or really any parameter at all). For example, to track Google Ad's just check for the existence of the parameter "gclid".

Pros:

- Easy to implement and get going - no code changes required
- Will track visitors based on originating source - even if they navigate around on your site before submitting the form

Cons:

- Potentially a lot of work in maintaining your audiences to match all your different campaigns if you want to track very granular on campaign ID's
- Adds 1-2 cookies that you'll need to declare
- Adds additional audiences (if you already have a lot)
- The form field will list all audiences - also those not related to UTM parameters so you'll have to do some post-processing on your data to clean them out
- Might not scale well, with many visitor groups to evaluate

Option 2: Simple code solution

The next option is in someways quite similar to the above one - in that it uses the Hidden Visitor Profile form element. But in this case we'll simply add a custom datasource to it, that will provide add the UTM parameters to each their own form field - if they exist.

Adding datasources to the Hidden Visitor Profile is quite easy - and something I've done in the past. You simply register a class that implements IVisitorDataSource, that checks the request and adds the value of the query parameters if they are set.

Name * ⓘ

Visible to
 Everyone

Languages
 en

ID, Type
 112, Hidden visitor profiling

[Content](#) [Settings](#)

Auto-collect visitor data ⓘ

Geo
 Profile
 User agent
 Audiences
 UTM Parameters

UTM Source

And so on - for each parameter:

Form elements
 Only Form elements, Form step, and Rich text (with placeholders) types.

...
 ...
 ...
 ...
 Submit button ...

You can drop form elements here or

And this way we get form submissions like this:

Form submissions				
Search		All columns	▼	From date
<input type="checkbox"/> Finalized Only				
Search				
<input type="checkbox"/>	Email	UTMSource	UTMMedium	Audience
<input type="checkbox"/>	test@test.dk	facebook	image	Submitted from Alloy Plan
<input type="checkbox"/>	test@test.dk	Facebook	Medium:Facebo...	Alloy Plan

Pros:

- Get exactly the data you want in individual columns
- Easy to extend to other data
- Easy to filter/use the data afterwards
- No cookies required

Cons:

- Only works if the form is on the page you land on with the query line parameters. Won't show up after you navigate the site

If you want to try it out, the code is in the code sample below.

```
using EPiServer.Forms.Core.VisitorData;
using EPiServer.ServiceLocation;
using Microsoft.Extensions.Primitives;

namespace FormsFun.Business.Forms;

[ServiceConfiguration(ServiceType = typeof(IVisitorDataSource))]
public class UTMParametersDataSource(IHttpContextAccessor httpContextAccessor, ILogger<UTMParametersDataSource> logger)
    //Remember to add services.AddHttpContextAccessor() in your startup / DI registration.
{
    private readonly IHttpContextAccessor _httpContextAccessor = httpContextAccessor ?? throw new ArgumentException("httpContextAccessor");
    private readonly ILogger _logger = logger ?? throw new ArgumentNullException(nameof(logger));

    public void SetData(ref object visitorData, string property)
    {
        var ctx = _httpContextAccessor?.HttpContext;
        if (ctx != null)
        {
            if (ctx.Request.Query[property] != StringValues.Empty)
            {
                visitorData = ctx.Request.Query[property].ToString();
                return;
            }
        }
        else
        {
            _logger.LogWarning("Unable to get HttpContext when trying to get UTM parameters for form");
        }
        visitorData = null;
    }

    public bool ClientSideMode => false;

    public string DataSourceName => "UTM Parameters";

    public IEnumerable<KeyValuePair<string, string>> GetProperties()
    {
        List<KeyValuePair<string, string>> rt = new List<KeyValuePair<string, string>>();
        rt.Add(new KeyValuePair<string, string>("utm_source", "UTM Source"));
        rt.Add(new KeyValuePair<string, string>("utm_medium", "UTM Medium"));
        rt.Add(new KeyValuePair<string, string>("utm_campaign", "UTM Campaign"));
        rt.Add(new KeyValuePair<string, string>("utm_term", "UTM Term"));
        return rt;
    }

    public async Task<object> SetDataAsync(string property)
    {
        object visitorData = null;
        SetData(ref visitorData, property);
        return visitorData;
    }
}
```

Optimizely (Episerver) CMS Website Improvements Tips and Tricks C#

CodeArt ApS

Teknikerbyen 5, 2830 Virum, Denmark

Email: info@codeart.dk

Phone: +45 26 13 66 96

CVR: 39680688



Copyright © 2025