

Javascript-based method to capture submissions: Method A (formcap.js)

We offer a small script which will capture form submissions for custom HTML-based webforms and then pass along the submitted data to Act-On by programmatically scanning for submissions to your selected form. Form submissions will be captured only after they have passed validation if your validation code is client-side.

Advantages:

- + For most forms already writing submission data to an external database, this is **the easiest method** and requires no adjustment of your existing form (other than adding the script somewhere within your webpage). It is designed to also only function after your custom form validation has occurred.
- + Will pass form submitters to the post-submit response page that is defined within your own systems (the post-submit redirect is not handled by Act-On, unlike the action URL method)

Disadvantages:

- May not work with forms that use "onclick" functionality within the submit button element
- Forms with server-side verification are not compatible
- Forms that utilize AJAX or have multiple submit events are not compatible
- **This script is very unlikely to work with multi-page forms, payment forms, login forms, or membership forms**

Step 1:

For each form this is used with, we need an equivalent "mirror" version of the Form to be created within Act-On. For example, if your custom form on your website captures First Name, Last Name, and Email, you'll need to create a form within Act-On that also requests these fields. Your website visitors will never see this Act-On variant of your form -- we simply need to receive your custom form's data.

Step 2:

Next, you'll want to make sure the input names match so that information can be passed properly. In the form builder, you'll see an "Edit" button appear when mousing-over any form field. Click "Edit", and take a look at "Data Field Name" on the resulting pop-up menu -- you want to ensure that your "Data Field Name" matches your custom form's HTML input name. For example, if your First Name field in Act-On has a Data Field Name of "fname", you'll want your custom form HTML to contain "<input name='fname'... " for the equivalent field. If you make any changes, be sure to click "next" twice in the Act-On form builder until you get to the last step, where you can save your form.

Step 3:

You'll want to grab your Act-On form's public URL. This is located under **Content > Forms > mouse-over your form > More > Get Public URL**

Step 4:

When you have your specific URL, you'll want to grab three elements from the form's public URL to plug into the script below:

- Your server domain and subdomain if applicable (such as actonsoftware.com, ci30.actonsoftware.com, or marketing.companyname.com if you use a custom URL for your account -- we do not need any http or www before the domain, or any characters after your ".com" or ".org")
- AID (account ID -- this will never change for your account)
- FID (form ID -- this will iterate for each unique form you have)
- DID (data ID -- this will most likely not change for you)

Your form's public URL will look like this. See the areas below in square brackets that contain the AID, FID, and DID:

[http://\[server\]/acton/form/\[aid\]/\[fid\]:\[did\]/0/index.htm](http://[server]/acton/form/[aid]/[fid]:[did]/0/index.htm)

and then fill it into the script below

Step 5:

Please take a look the following script. However, before it is embedded this on your webpage, you'll want to ensure the www, xxxx, yyyy, zzzz, and server elements are configured below

```
<script type="text/javascript">
var aoProtocol = location.protocol;
if ( aoProtocol.indexOf('http') < 0 ) aoProtocol = 'http:.';
var aoCAP = {
aid: 'www',
fid: 'xxxx',
did: 'd-yyyy',
server: 'your_subdomain_and_domain_here_without_<a href="http://www">http://www',
formName: 'zzzz',
protocol: aoProtocol
};
document.write( '<script type="text/javascript"
src="'+aoCAP.protocol+'/' +aoCAP.server+'/acton/js/formcap.min.js"><'+'/script>' );
</script>
```

Above: Replace www with your AID, xxxx with your FID, and yyyy with your DID from your Act-On form's custom URL. These three elements are all present in your Act-On form's public URL, as shown in step 4.

Step 6:

Lastly, replace zzzz above with your custom form's name. This particular element is not from within your Act-On account -- it is present within your custom form's HTML. For example, if you view the source of your form you should see something similar to `<form name="formNameExample">`.

Please note you can change "formName" to "formId" to use your custom form's ID attribute, if instead you see `<form ID="formIdExample">`

Formcap.js frequently asked questions:

What if I don't see a id or name attribute inside the <form> tag?

This script requires a `name="nameExample"` or `id="idExample"` inside your `<form>` tag (within your custom form's HTML) in order to be able to target your form. To successfully integrate with this Javascript-based external post method, you will need to add one of these attributes to your form's HTML inside the `<form>` tag.

Where should I place this script?

It can be placed anywhere on your page where Javascript can run (head or body), though we advise placing it at the foot of your page.

What if I place this script on a page that doesn't contain the correct form?

The external formcap.js script will load, but it will take no action and there should not be any detrimental effect on your webpage

Can I use this multiple times on the same page because I have more than one form per webpage?

There is no harm in using multiple formcap.js scripts on the same webpage.

One function of this script is that it will load the full formcap.js external script located at <http://www.yoursubdomain.domain.com/acton/js/formcap.js> on your page once per page-load, so multiple instances of the above script will load that external script multiple times. If you want to save the few Kilobytes of redundancy this would create, all you need to do is omit the document.write line from the extra instances of the script to avoid this redundant file loading.

Is "formId" or "formName" case-sensitive?

Yes -- these properties are case-sensitive because Javascript is case-sensitive. These are properties of the script itself, and don't correspond to your form's HTML capitalization. For example, you can use `<form id="myID">` or `<form ID="myID">` or `<form name="myName">` or `<form NAME="myName">` and so on.

I'm not seeing any submissions appear in the signup list in my Act-On account. What should I do?

Go to the URL with your custom form, submit the form, and check the Developer tools or Javascript console in your browser. Most often, the source of the problem is a syntax error such as a missing single-quote. The developer interface of your browser should highlight any script

errors it catches. If it still is not functioning or you see an unrelated error, please contact support@actonsoftware.com.

- Solutions to some common console errors:
 - *"SyntaxError: Expected token '}'"* is caused by a missing quote. Please ensure all single quotes or double quotes are properly paired (every opening quote needs a closing quote of the same type). This also may be caused by a [smart quote](#), which should be replaced with a single quote.
 - *"Property 'submit' of object #<HTMLFormElement> is not a function"* is caused by your form's submit button having a name attribute of "submit". Please avoid using "submit" as your form input names.

Completed examples:

Simple custom form HTML with script:

```
<!-- EXAMPLE PUBLIC URL FOR ACT-ON FORM, built using the form builder -->
http://www.marketing.actomatic.com/acton/form/1234/0002:d-0001/0/in...
<!-- simple EXTERNAL form example that would be hosted elsewhere -->
<form name='contactUs' onsubmit='SubmitToMyCustomDatabaseOutsideOfActOn();'>
<!-- remember to replace the action URL in this example with your own, obtained through
Reports > Forms and Media -->
<input name='firstname' type='text' placeholder='First Name'>
<input name='lastname' type='text' placeholder='Last Name'>
<input name='email' type='text' placeholder='Email'>
<input type='submit' value='submit' >
</form>
<!-- script below to place on the webpage containing your custom external form -->
<script>
var aoProtocol = location.protocol;
if ( aoProtocol.indexOf('http') < 0 ) aoProtocol = 'http:';
var aoCAP = {
aid: '1234',
fid: '0002',
did: 'd-0001',
server: 'marketing.actomatic.com',
formName: 'contactUs',
protocol: aoProtocol,
};
document.write( '<script type="text/javascript"
src="'+aoCAP.protocol+'/'+'+aoCAP.server+'/acton/js/formcap.min.js"><'+'/script>' );
</script>
```

Simple custom form HTML with script:

```
<!-- EXAMPLE PUBLIC URL FOR ACT-ON FORM, built using the form builder -->
```

<http://www.ci30.actonsoftware.com/acton/form/4321/003b:d-0001/0/ind...>

```
<!-- simple EXTERNAL form example that would be hosted elsewhere -->
<form ID='freeTrial' onsubmit='SubmitToMyCustomDatabaseOutsideOfActOn();'>
<input name="firstname" type="text" placeholder='First Name'>
<input name='lastname' type='text' placeholder='Last Name'>
<input name='email' type='text' placeholder='Email'>
<input type='submit' value='submit' >
</form>
```

```
<!-- script below to place on the webpage containing your custom external form -->
<script>
var aoProtocol = location.protocol;
if ( aoProtocol.indexOf('http') < 0 ) aoProtocol = 'http:';
var aoCAP = {
aid: '4321',
fid: '003b',
did: 'd-0001',
server: 'ci30.actonsoftware.com',
formId: 'freeTrial',
protocol: aoProtocol,
};
document.write( '<script type="text/javascript"
src="'+aoCAP.protocol+'/' +aoCAP.server+'/acton/js/formcap.min.js"
<'+'/script>' );
</script>
```