

Cognito Form: Embed script implementation

First, make sure all the UTM fields are created in Cognito Form. Each field should be separate and Field Name should match the name you will use later in the implementation (see the image below).

The image shows the Cognito Form editor interface. On the left, the 'Field Settings - Textbox' panel is visible, containing the following fields:

- Label:** A text input field containing 'utm campaign'.
- Field Name:** A text input field containing 'Utmcampaign'.
- Type:** Three radio button options: 'Single Line' (selected), 'Multiple Lines', and 'Password'.
- Placeholder Text:** A text input field containing 'test'.
- Format Validation:** A dropdown menu with a downward arrow.
- Default Value:** A text input field (partially visible).

On the right, a live preview of the form is shown. It includes a title 'test', a label 'testing', an 'Email' field with an envelope icon, and a 'utm campaign' field with a text input containing 'A test'. A context menu is open over the 'utm campaign' field, showing options: Cut, Copy, Delete, Insert Field, Make Smaller, Make Bigger, and Justify Row. Below the form fields is a 'Submit' button and a message: 'Thank you for filling out the form. Your response has been recorded.'

Your default embed code from Cognito Form should look like this

```
<script src="https://services.cognitoforms.com/s/<YOUR FORM ID>"></script>
<script>Cognito.load("forms", { id: "2"});</script>
```

Modify the embed code from Cognito Forms like the following. Notice that, we are adding entry fields to the `Cognito.load`.

```
<script src="https://services.cognitoforms.com/s/<YOUR FORM ID>"></script>
<script>Cognito.load("forms", { id: "2",
entry: {
  ["UTMCampaign":Cookies.get("utm_campaign"),
    "UTMSource":Cookies.get("utm_source") ,
```

```
"UTMMedium":Cookies.get("utm_medium"),
"UTMContent":Cookies.get("utm_content"),
"UTMTerm":Cookies.get("utm_term"),
"IP": Cookies.get("handl_ip"),    "Organic": Cookies.get("organic_source_str")
});</script>
```

Here we only illustrated `utm_*` and `handl_ip`. However you can use all the other parameters very similar way. See the list of all the parameters available [here](#)

Native WP Shortcodes

NOTE: Cognito does not support hidden field in the form out of the box. However you can make a regular text field hidden by creating a condition that never satisfies like this.

The screenshot displays a form configuration interface with a modal window titled "Visible When...". The modal is used to define conditions for when a field should be visible. It features a list of conditions, each with a field name, an operator, and a value. The first condition is "Email" with the operator "contains" and the value "123131312asdsad". Below this, there is a section for adding more conditions, labeled with a plus sign and the word "or". The modal also includes a "Basic Editor" tab, an "Advanced Editor" tab, and buttons for "Cancel" and "Save".

Placeholder Text
test

Format Validation

Default Value

Number of Characters
Minimum to Maximum

Help Text

Show This Field
When Email is filled out

Require This Field
Always When Never

Read-Only
Always When Internally Never

Limit Quantity
No Duplicates Specific Quantity Never

Show Custom Error
When Never

Submit

Visible When...

Email contains 123131312asdsad + and

+ or

Basic Editor Advanced Editor

Cancel Save

Revision #8

Created Fri, Oct 23, 2020 4:27 PM by [Leman](#)

Updated Wed, Jan 17, 2024 3:12 AM by [Leman](#)