

# Cognito Forms Integration

HandL UTM Grabber / Tracker How to collect and track UTM variables via Cognito Forms step by step

- [Cognito Form: Embed script implementation](#)
- [Cognito Form: Capture UTMs using Iframe](#)

# 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 configuration interface. On the left, the 'Field Settings - Textbox' panel is open, showing the following settings:

- Label:** utm campaign
- Field Name:** Utmcampaign
- Type:** Single Line (selected), Multiple Lines, Password
- Placeholder Text:** test
- Format Validation:** (empty dropdown)
- Default Value:** (empty field)

On the right, a preview of the form is shown. It includes a title 'test', a label 'testing', an 'Email' field, and a 'utm campaign' field with the placeholder text 'A test'. A context menu is visible 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 image shows a form builder interface with a modal dialog titled "Visible When...". The modal contains a condition: "Email" contains "123131312asdsad". The background shows various form field settings:

- 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**

The modal dialog has a "Basic Editor" tab selected and a "Save" button.

# Cognito Form: Capture UTMs using Iframe

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 interface. On the left is the 'Field Settings - Textbox' panel. It includes fields for 'Label' (utm campaign), 'Field Name' (Utmcampaign), 'Type' (Single Line selected), 'Placeholder Text' (test), and 'Format Validation'. On the right is a preview of the form. It has a title 'test', a label 'testing', an 'Email' field, and a 'utm campaign' field. A context menu is open over the 'utm campaign' field, showing options like 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 iframe code from Cognito Form should look like this

```
<iframe src="https://www.cognitoforms.com/f/<YOUR FORM ID>?id=2"
style="position:relative;width:1px;min-width:100%;*width:100%;" frameborder="0"
scrolling="yes" seamless="seamless" height="323" width="100%"></iframe><script
src="https://www.cognitoforms.com/scripts/embed.js"></script>
```

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

```
<iframe src="https://www.cognitoforms.com/f/<YOUR FORM ID>?id=2"
style="position:relative;width:1px;min-width:100%;*width:100%;" frameborder="0"
```

```
scrolling="yes" seamless="seamless" height="323" width="100%"></iframe><script
src="https://www.cognitoforms.com/scripts/embed.js"></script>
<script>
Cognito.prefill({
  "Utmcampaign": "[utm_campaign]",
    "UTMSource": "[utm_source]" ,
    "UTMMedium": "[utm_medium]",
    "UTMContent": "[utm_content]",
    "UTMTerm": "[utm_term]",
    "IP": "[handl_ip]",
    "Organic": "[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 the Cognito Forms editor interface. A modal window titled "Visible When..." is open, showing a conditional logic rule. The rule is configured as follows:

- Field: Email
- Operator: contains
- Value: 123131312asdsad
- Logic: and

The modal also includes a "Basic Editor" tab, an "Advanced Editor" tab, and "Cancel" and "Save" buttons. In the background, the form field configuration panel is visible, showing options for Placeholder Text, Format Validation, Default Value, Number of Characters, 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), and Show Custom Error (When, Never).