Forms Editor Block

The Forms Editor block is used to build HTML forms and present them anywhere within your site. When site visitors complete your form, Titan CMS can email the data to one or more designated email addresses and store the submitted information in our database. It can also track your form responses through Titan CMS or download the data as a spreadsheet.

Features:

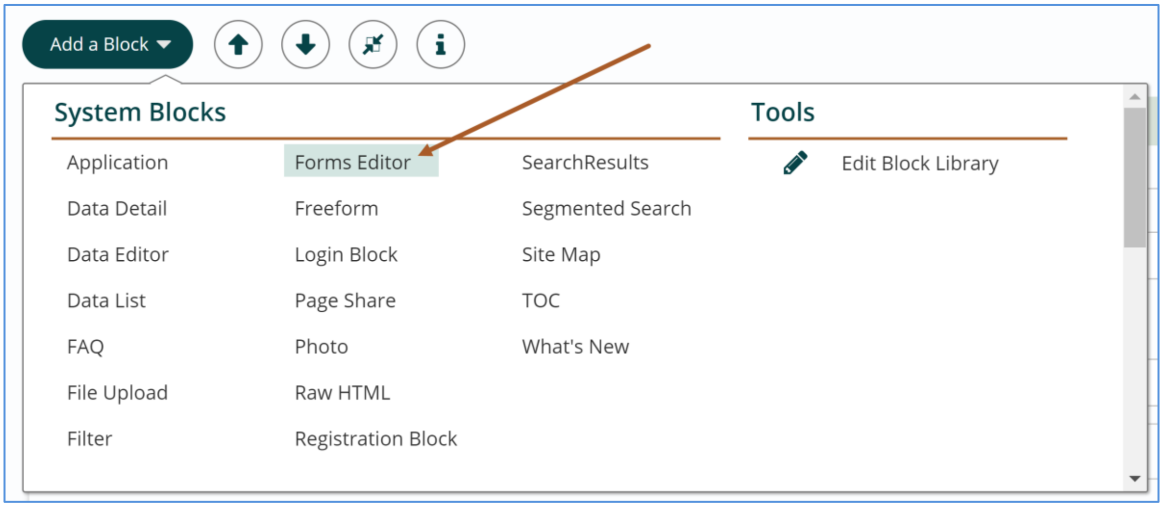
- Allows simple and complex field validation rules
- Presents a textual confirmation message after form submissions
- Email support for form data recipients with selection of desired fields
- Optional email support for form submitters with selection of desired fields
- Results retrieval interface for data analysis
- Editor functionality

Common Uses:

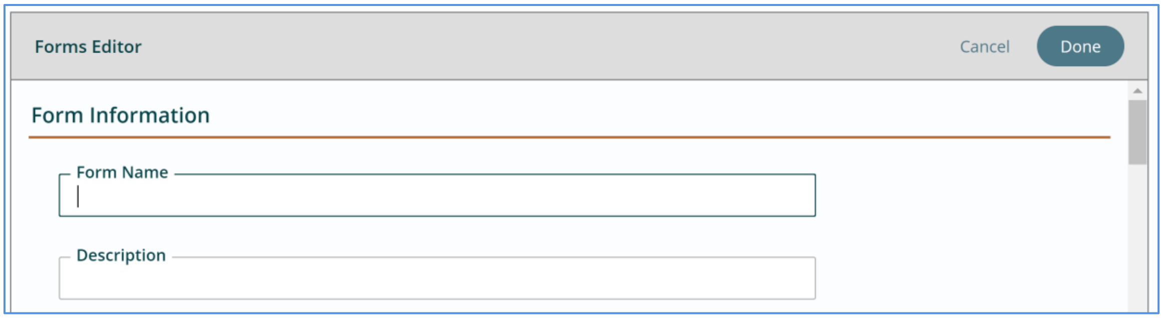
- Event Registration forms
- "Contact Us" style forms
- Requests for information

Create a Basic Form With Text Fields

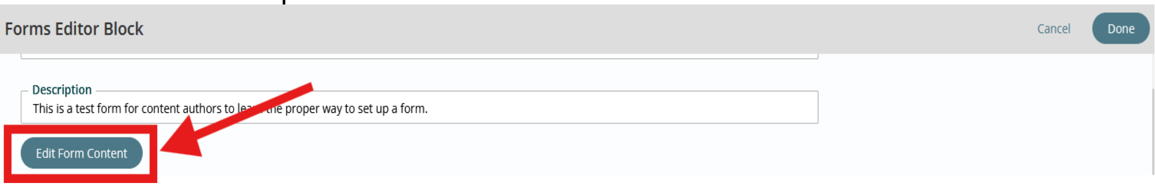
1. Open a page.
2. From the Add A Block menu, select Forms Editor.



3. Give your form a name and description. The description will not be visible anywhere on the site.

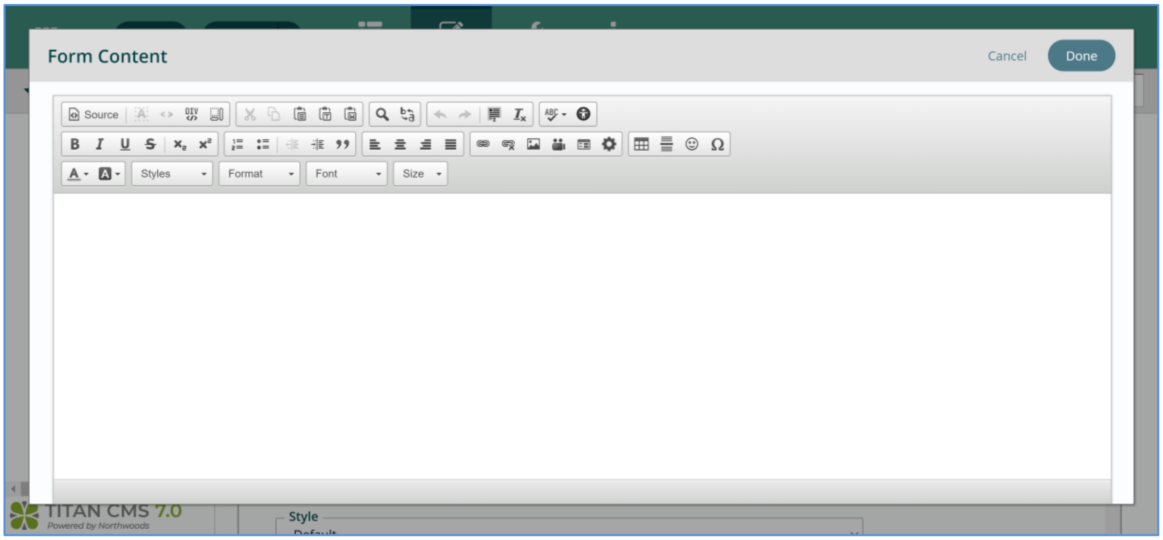


4. Click the Edit link to open the Forms Editor block Editor.



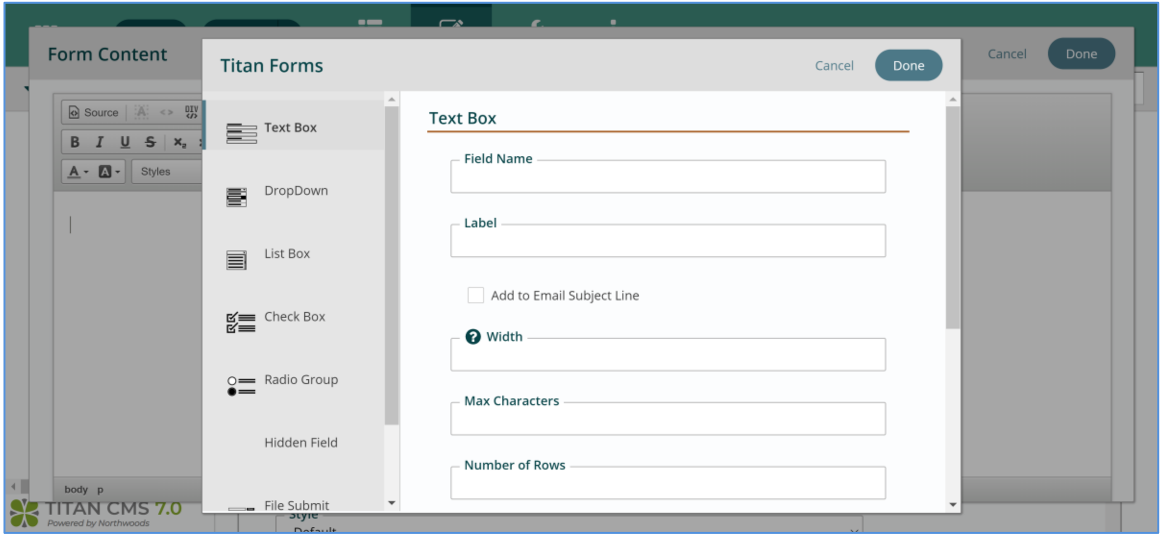
Description: This is a test form for content authors to learn the proper way to set up a form.

5. You will now see the Form Content editor window, which is very similar to the Freeform Block. You can enter regular text here as you would in a freeform block, as well as form fields.



6. There is a Titan Forms button in your Forms Editor. To add in input type to the form, click the Forms Editor Block icon ☰.

7. Upon selecting Titan Forms, the following dialog box appears:

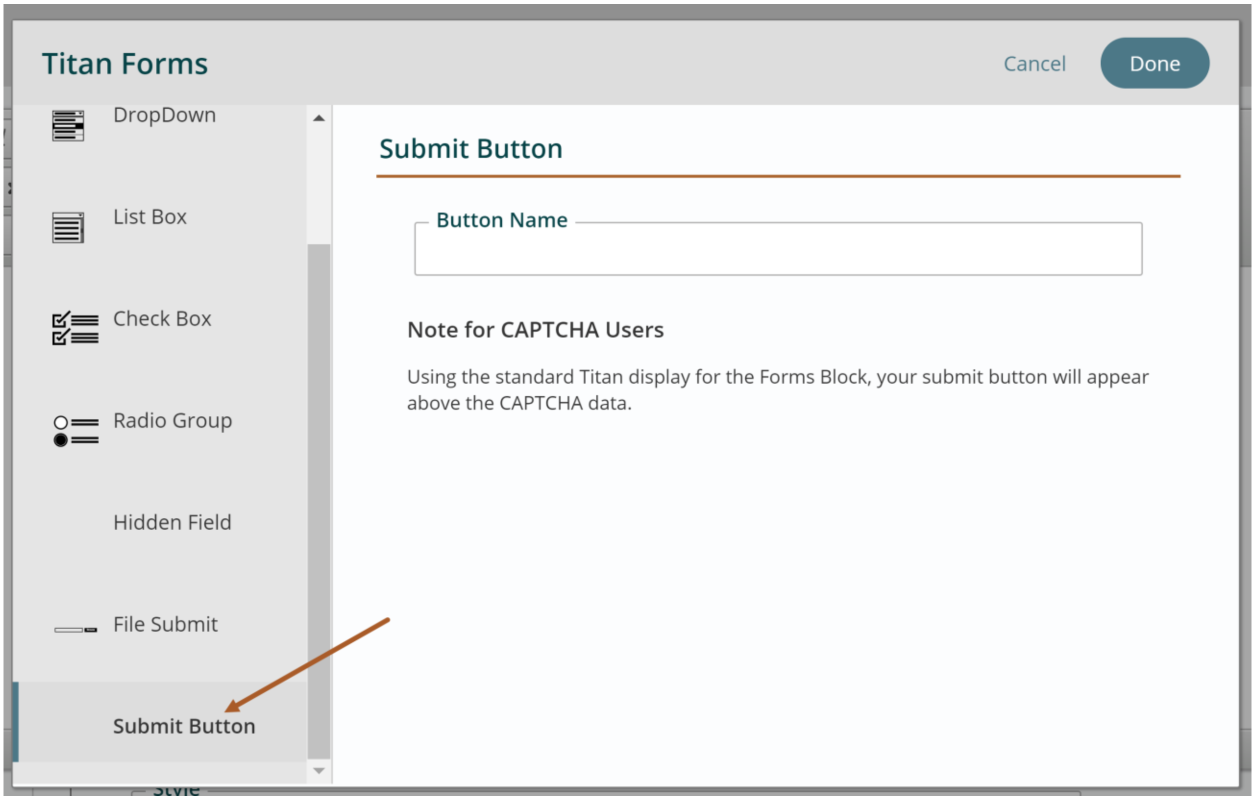


8. To enter a basic Text Box field (default display):
   a. **Field Name:** This becomes the internal database name for this field. It cannot have any spaces or special characters. Example: FieldName or Email
   b. **Label:** This will appear before your form field both in the workstation and on the display site. If a field is required, best practices for accessibility suggest including (required) at the end of this field to ensure this is clearly understood visually and by screen reader users. ***NEW IN TITAN CMS 7.0: You don't need to enter labels separately in the Freeform editor!***
   c. **Pixel Width/Max Characters:** You can leave this at the default. Our styling overrides this field.
   d. **Required field:** If you want the user to complete this field and receive a message if it is left blank, check this box.
   e. **Error Message:** Enter the error message you wish to display if there is a problem with the field entry. Error messages must indicate which field is in error. For example, "Email is a required field."
   f. **Validation Type:** Use this for Email or Phone numbers. Titan will verify the formatting of the field, if selected.
   g. Click DONE when you're done.

9. Continue to build your form, adding fields for the information you would like to collect. See Additional Form Field Options to read about other field options.

10. Scroll down in the Titan Forms Editor to find the **Submit** option. For simple forms with button text "Submit", it is not necessary to add a submit button. Use this if you would like to use different text on the button or have another need that requires customization to

the default button.



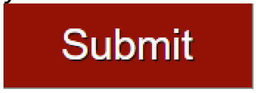
Submit Button

Button Name

Note for CAPTCHA Users

Using the standard Titan display for the Forms Block, your submit button will appear above the CAPTCHA data.

11. Whatever text you put in the Button Name field will appear as the submission button for your form:

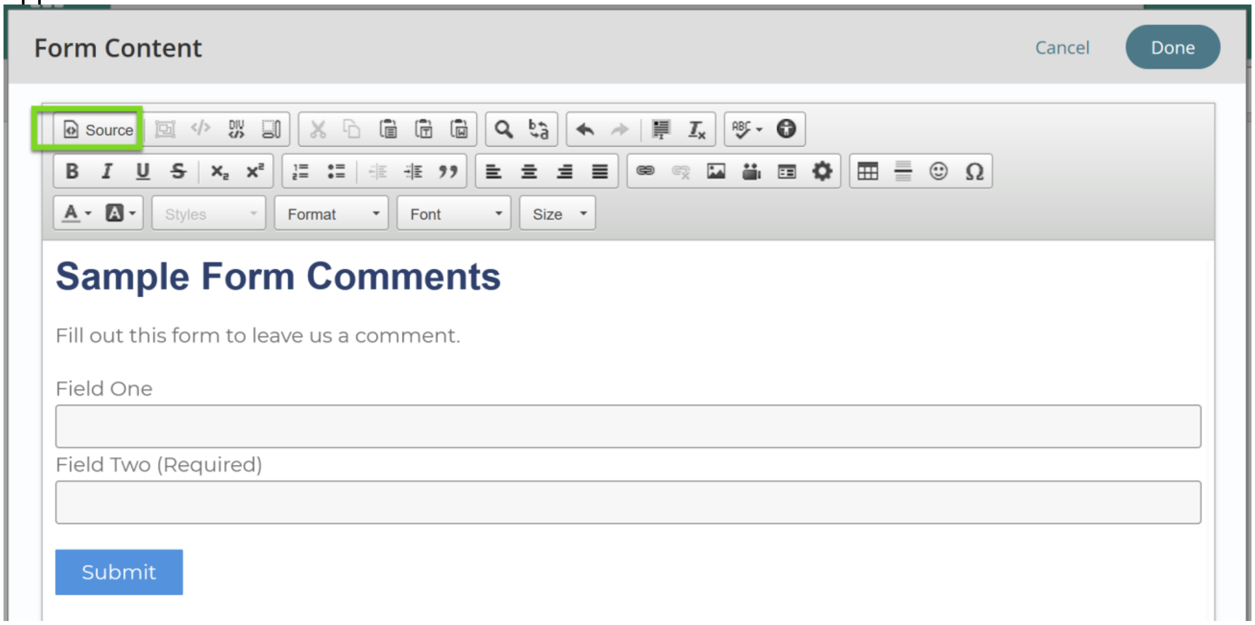


12. Click Done in the Form Content Editor. This will bring you back to the main Forms Editor. Hit Done there as well.
13. Publish your page. For thank you messages and other follow-up text or page redirects after the form is submitted, see “Post Submit – User View.”

Accessibility Edits in the Source Code

In certain cases, you will want to go into the source code to update the form and make it more accessible for screen reader users. Common reasons for this would be if you have instructions for the form, required fields, or multiple choice options like checkboxes and radio buttons that need descriptions for the group of options.

1. After adding the necessary inputs using the methods above, hit the Source button in the upper left corner of the Form Editor.



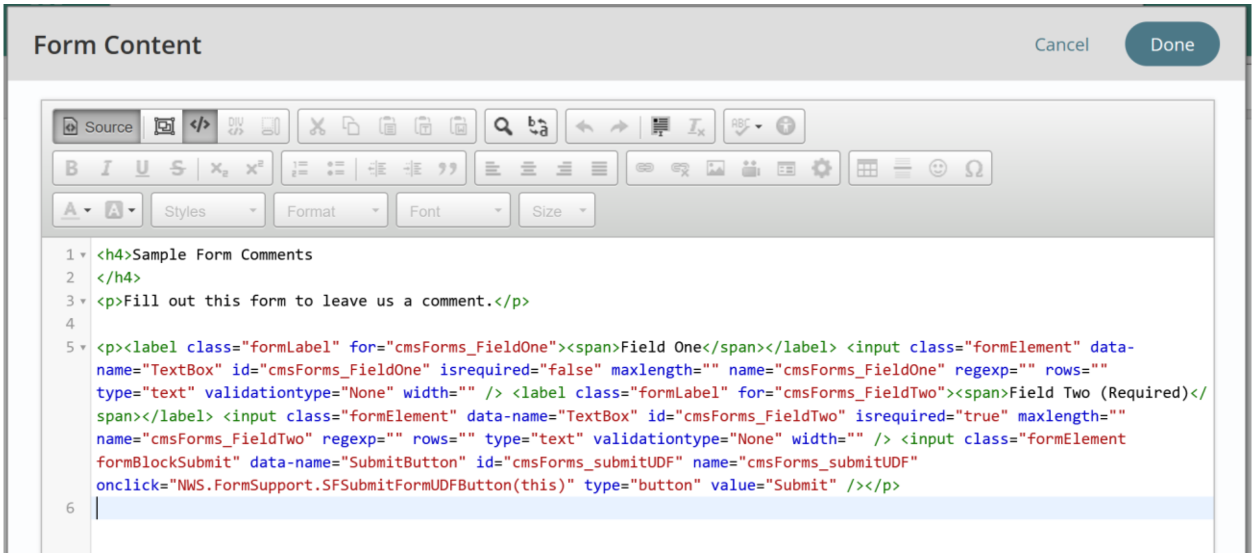
Sample Form Comments

Fill out this form to leave us a comment.

Field One

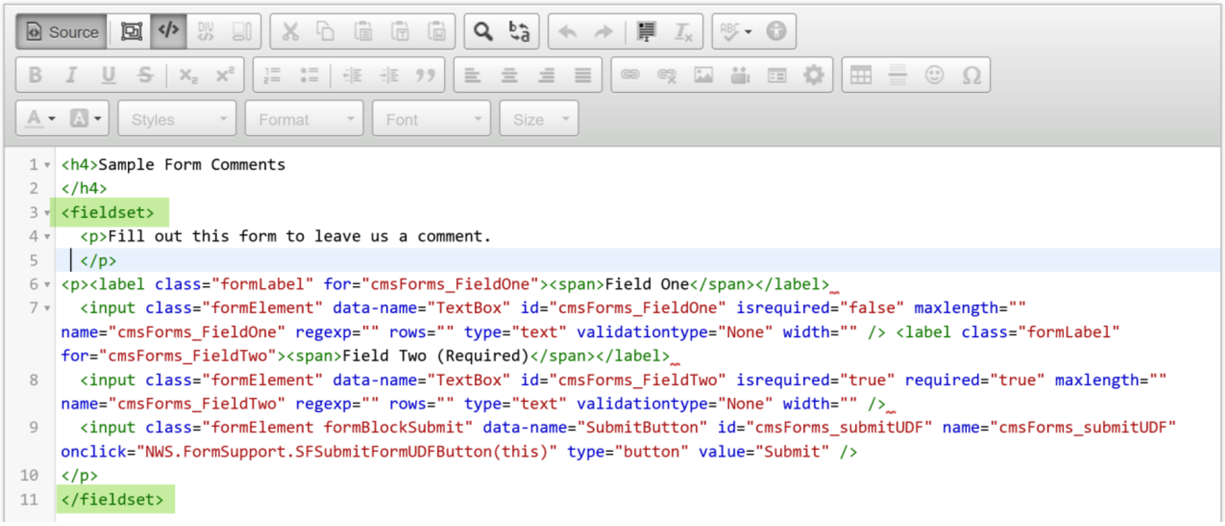
Field Two (Required)

2. The source code for the above form would like:



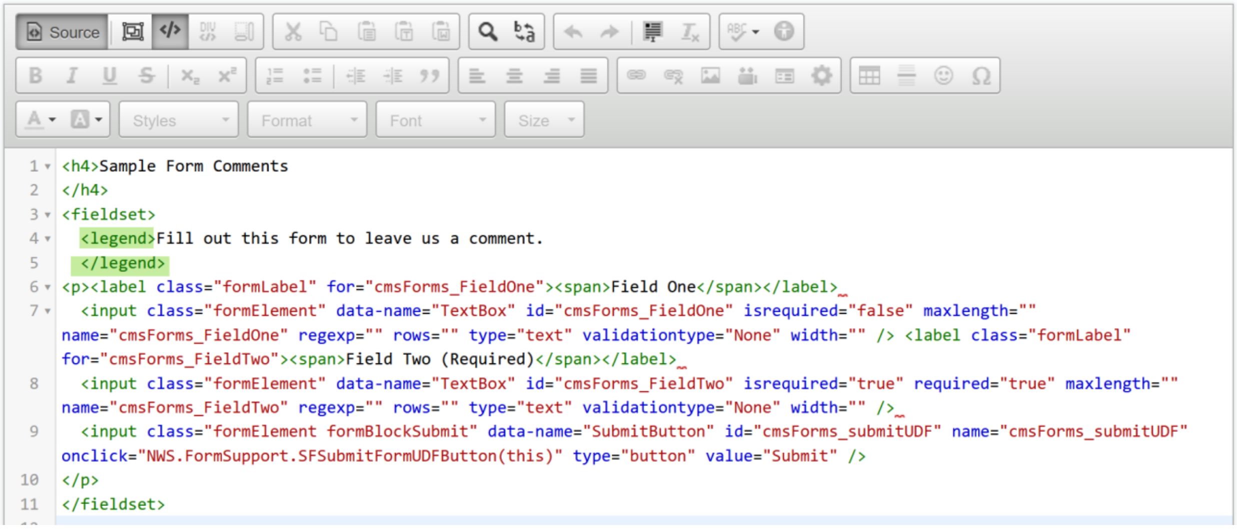
```
<h4>Sample Form Comments
</h4>
<p>Fill out this form to leave us a comment.</p>

<p><label class="formLabel" for="cmsForms_FieldOne"><span>Field One</span></label> <input class="formElement" data-
name="TextBox" id="cmsForms_FieldOne" isrequired="false" maxlength="" name="cmsForms_FieldOne" regexp="" rows=""
type="text" validationtype="None" width="" /> <label class="formLabel" for="cmsForms_FieldTwo"><span>Field Two (Required)</
span></label> <input class="formElement" data-name="TextBox" id="cmsForms_FieldTwo" isrequired="true" maxlength=""
name="cmsForms_FieldTwo" regexp="" rows="" type="text" validationtype="None" width="" /> <input class="formElement
formBlockSubmit" data-name="SubmitButton" id="cmsForms_submitUDF" name="cmsForms_submitUDF"
onclick="NWS.FormSupport.SFSubmitFormUDFButton(this)" type="button" value="Submit" /></p>

```

3. If you have instructions for the form as a whole that the user should understand before beginning, add a <fieldset> element around the entirety of the instructions and the inputs you created . Insert “<fieldset>” at the start of the form content, the editor should automatically create a closing </fieldset> tag which you will need to move to the end of the content you would like to wrap. What this does is mark the whole form as one associated collection.



```
<h4>Sample Form Comments
</h4>
<fieldset>
  <p>Fill out this form to leave us a comment.
  </p>
<p><label class="formLabel" for="cmsForms_FieldOne"><span>Field One</span></label>
  <input class="formElement" data-name="TextBox" id="cmsForms_FieldOne" isrequired="false" maxlength="" name="cmsForms_FieldOne" regexp="" rows="" type="text" validationtype="None" width="" /> <label class="formLabel" for="cmsForms_FieldTwo"><span>Field Two (Required)</span></label>
  <input class="formElement" data-name="TextBox" id="cmsForms_FieldTwo" isrequired="true" required="true" maxlength="" name="cmsForms_FieldTwo" regexp="" rows="" type="text" validationtype="None" width="" />
  <input class="formElement formBlockSubmit" data-name="SubmitButton" id="cmsForms_submitUDF" name="cmsForms_submitUDF" onclick="NWS.FormSupport.SFSubmitFormUDFButton(this)" type="button" value="Submit" />
</p>
</fieldset>
```

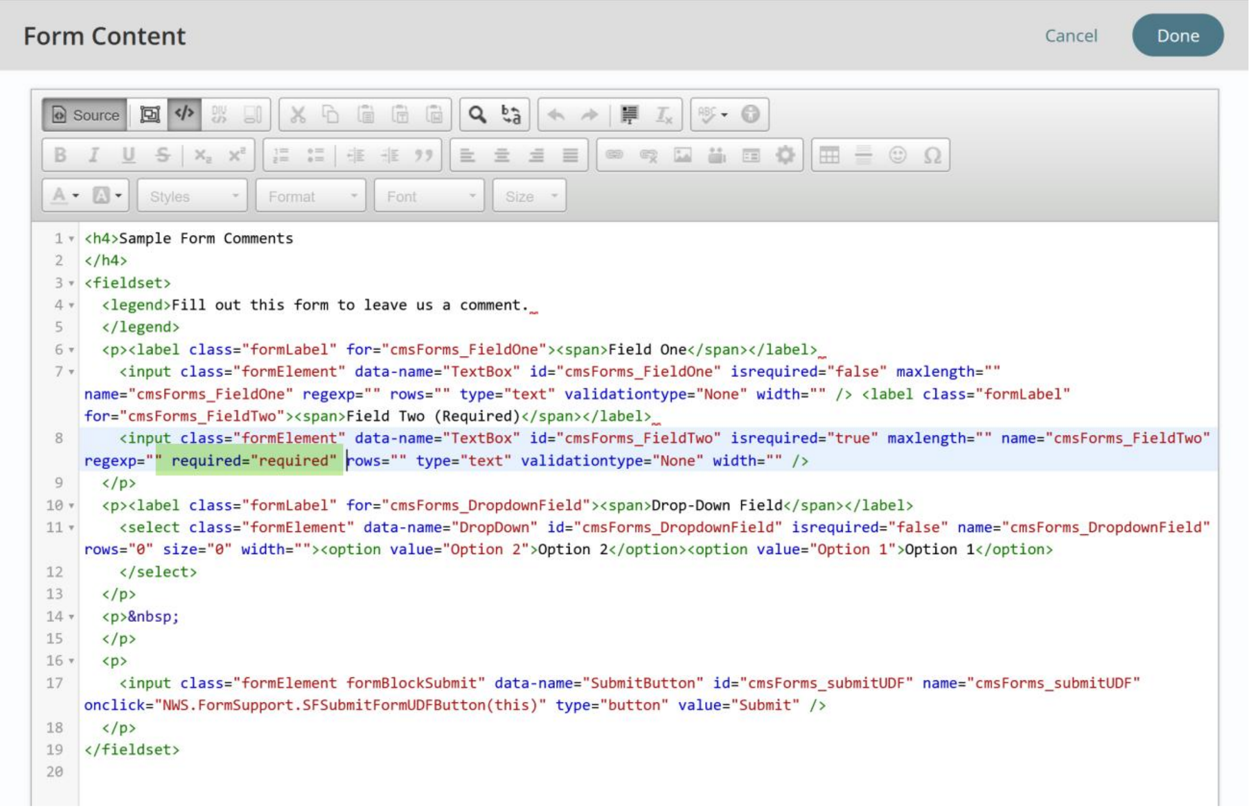
4. You can nest fieldset elements, and should do so if you have multiple choice options in your form. See the instructions for multiple choice fields to learn more.
5. If you have any instructions for your form, you will want them to be visible and read by assistive technology as a legend. Put <legend></legend> around the start and end of the instruction text. All instructions and examples should come before they are needed in the form, not as further instructions below or after a field.



```
<h4>Sample Form Comments
</h4>
<fieldset>
  <legend>Fill out this form to leave us a comment.
  </legend>
<p><label class="formLabel" for="cmsForms_FieldOne"><span>Field One</span></label>
  <input class="formElement" data-name="TextBox" id="cmsForms_FieldOne" isrequired="false" maxlength="" name="cmsForms_FieldOne" regexp="" rows="" type="text" validationtype="None" width="" /> <label class="formLabel" for="cmsForms_FieldTwo"><span>Field Two (Required)</span></label>
  <input class="formElement" data-name="TextBox" id="cmsForms_FieldTwo" isrequired="true" required="true" maxlength="" name="cmsForms_FieldTwo" regexp="" rows="" type="text" validationtype="None" width="" />
  <input class="formElement formBlockSubmit" data-name="SubmitButton" id="cmsForms_submitUDF" name="cmsForms_submitUDF" onclick="NWS.FormSupport.SFSubmitFormUDFButton(this)" type="button" value="Submit" />
</p>
</fieldset>
```

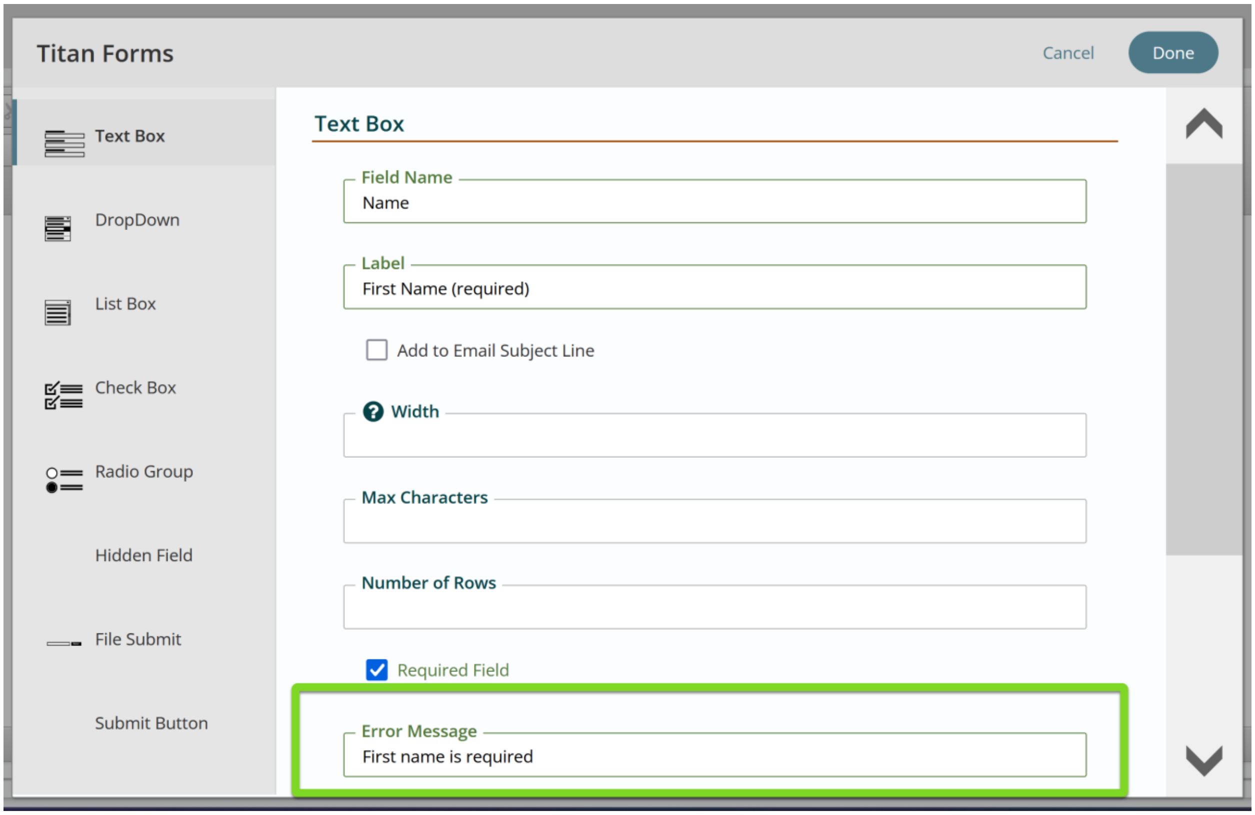
6.

7. If any of your fields are required, you will want to add in required="required" to the code.



```
<h4>Sample Form Comments
</h4>
<fieldset>
  <legend>Fill out this form to leave us a comment.
  </legend>
  <p><label class="formLabel" for="cmsForms_FieldOne"><span>Field One</span></label>
    <input class="formElement" data-name="TextBox" id="cmsForms_FieldOne" isrequired="false" maxlength="" name="cmsForms_FieldOne" regexp="" rows="" type="text" validationtype="None" width="" /> <label class="formLabel" for="cmsForms_FieldTwo"><span>Field Two (Required)</span></label>
    <input class="formElement" data-name="TextBox" id="cmsForms_FieldTwo" isrequired="true" maxlength="" name="cmsForms_FieldTwo" regexp="" required="required" rows="" type="text" validationtype="None" width="" />
  </p>
  <p><label class="formLabel" for="cmsForms_DropdownField"><span>Drop-Down Field</span></label>
    <select class="formElement" data-name="DropDown" id="cmsForms_DropdownField" isrequired="false" name="cmsForms_DropdownField" rows="0" size="0" width=""><option value="Option 2">Option 2</option><option value="Option 1">Option 1</option>
    </select>
  </p>
  <p>&nbsp;
  </p>
  <p>
    <input class="formElement formBlockSubmit" data-name="SubmitButton" id="cmsForms_submitUDF" name="cmsForms_submitUDF" onclick="NWS.FormSupport.SFSubmitFormUDFButton(this)" type="button" value="Submit" />
  </p>
</fieldset>
```

8. Additonally, all required fields need descriptive error messages. Error messages must be descriptive of the field in question, e.g. “First Name is required” instead of “this field is required.” Error messages are entered in the Error Message field of Titan Forms.



Text Box

Field Name: Name

Label: First Name (required)

□ Add to Email Subject Line

Width:

Max Characters:

Number of Rows:

☑ Required Field

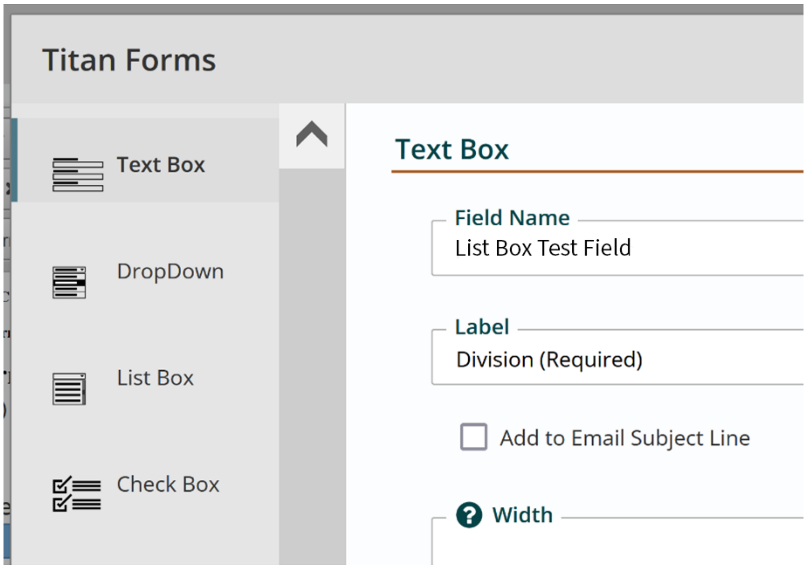
Error Message: First name is required

9. **A note about source edits:** When you enter labels directly in the source code, they do not always appear in the Titan Forms Editor. They do appear on the site, and function correctly. However, especially for fields other than text boxes and fields that contain a paragraph break, the Forms Editor does not always pull that in to populate the label field on the WYSIWYG side of the editor in Titan.

Edit a Form Field

1. Click into the field in the form content.
2. Click the Titan Forms icon.
3. The Form Content box will open. Make all desired changes.
4. Click Done to retain the changes.
5. Click Save or Publish from the App Ribbon.

Note: when you edit a field that is not a simple text box field, be sure to select the correct field type in the left-hand menu. The Text type field will be selected by default.



Text Box

Field Name: List Box Test Field

Label: Division (Required)

□ Add to Email Subject Line

Width

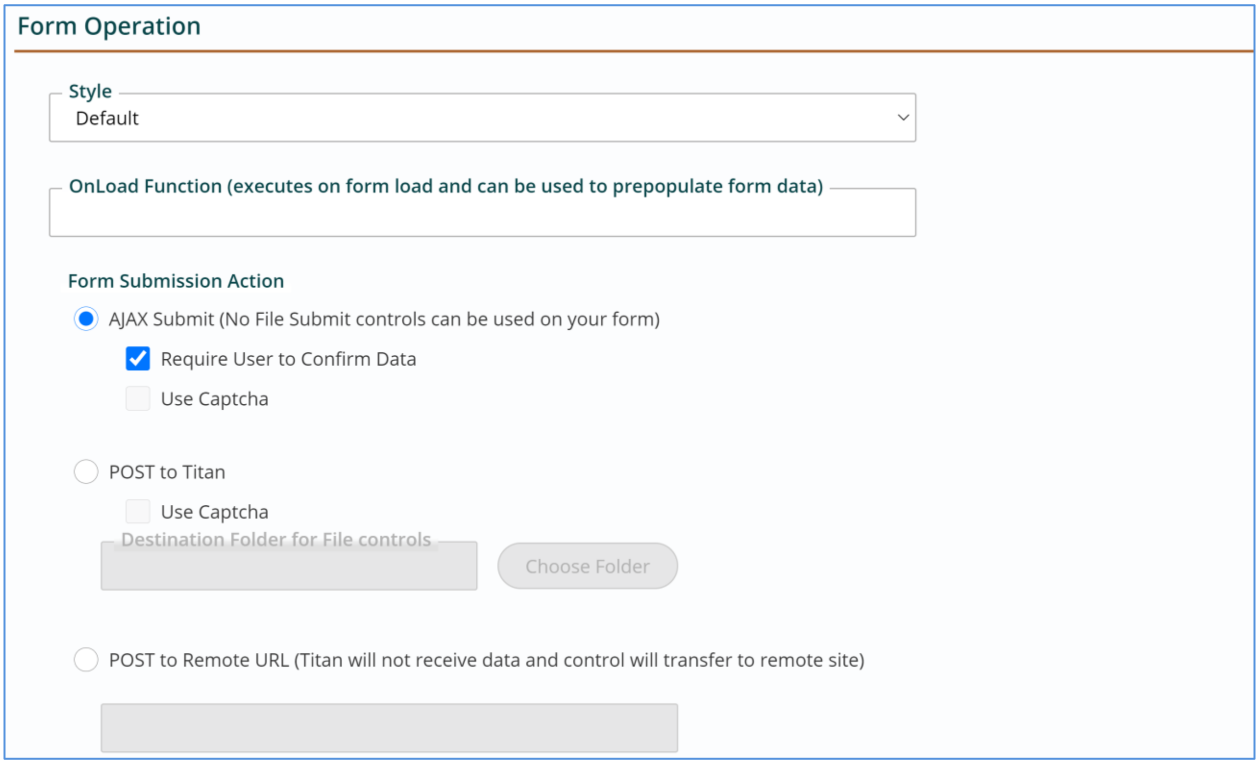
Note for editing in earlier versions of Titan:

- In Titan CMS Version 6.9 or earlier, when you edit a form field the Field Name will be grayed out, because this is a database field and not editable. If you need to change the name field, delete this check box and add a new check box.

Delete A Form Field

1. Click on the field in the form content.
2. Select delete on your keyboard. The field will be removed from your form.
3. Click Save or Publish from the App Ribbon.

Form Operation Functions



Form Operation

Style: Default

OnLoad Function (executes on form load and can be used to prepopulate form data)

Form Submission Action

● AJAX Submit (No File Submit controls can be used on your form)

- ☑ Require User to Confirm Data
- □ Use Captcha

○ POST to Titan

- □ Use Captcha
- Destination Folder for File controls

○ POST to Remote URL (Titan will not receive data and control will transfer to remote site)

Style

Generally, you will use the default style for your site. If your Style Guide has multiple form styles, you could choose an option from the drop down.

OnLoad Function

The OnLoad Function is used to pre-populate a text field in the form based on function. A function will need to be created. To enable this function, please contact your Account Director for developer assistance. More information can be found on the Titan CMS Documentation site.

Form Submission Action

AJAX Submit

- ***This is not available if you are using the file submit option***
- Require User to Confirm Data (User will use Send and Edit buttons when reviewing)

- Use Captcha. In order to use Captcha, you will need to configure Titan Administration with a captcha key

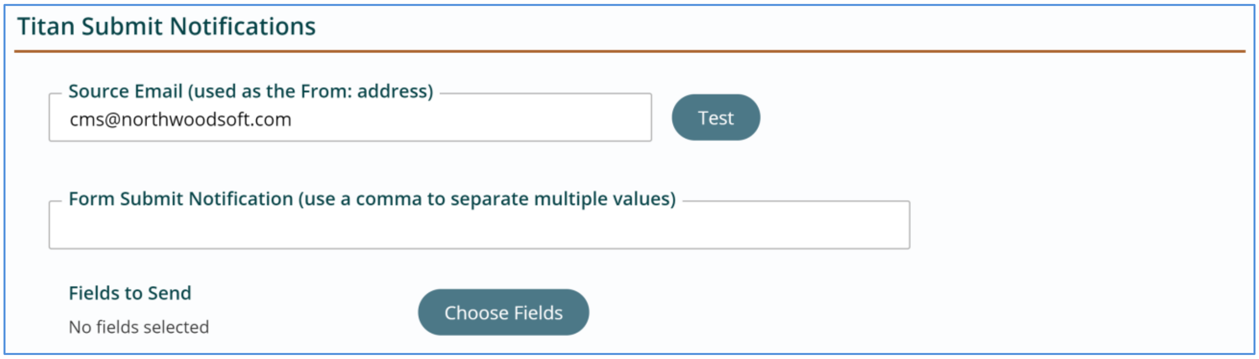
POST to Titan

- **Use Captcha -** In order to use Captcha, you will need to configure Titan Administration with a captcha key.
- **Choose Folder** - Use this option when File Submit is configured in your form. This is where you will select the folder where submitted files will be stored.
  NOTE: the folder must have Security set to Everyone, or the File Submit action won't work.

POST to Remote URL

- Titan CMS will not receive data and control will transfer to remote site.

Titan Submit Notifications



Titan Submit Notifications

Source Email (used as the From: address): cms@northwoodsoft.com

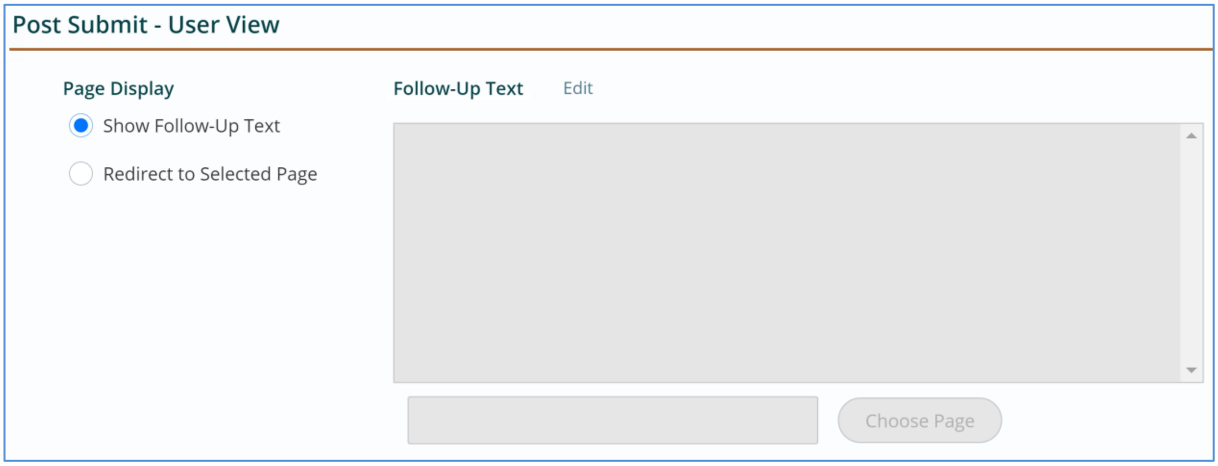
Form Submit Notification (use a comma to separate multiple values):

Fields to Send
No fields selected

1. Under Titan Submit notifications, set your Source Email to an organization/business address.
2. The Form Submit Notification should be the email address of the individual receiving the submission. You can have multiple addresses, but they must be separated by commas.
3. Click the Choose Fields button and select the fields you wish to send in the notification email.

Post Submit – User View

Add follow up text to page display following form submission (Page Display)



Post Submit - User View

Page Display

● Show Follow-Up Text

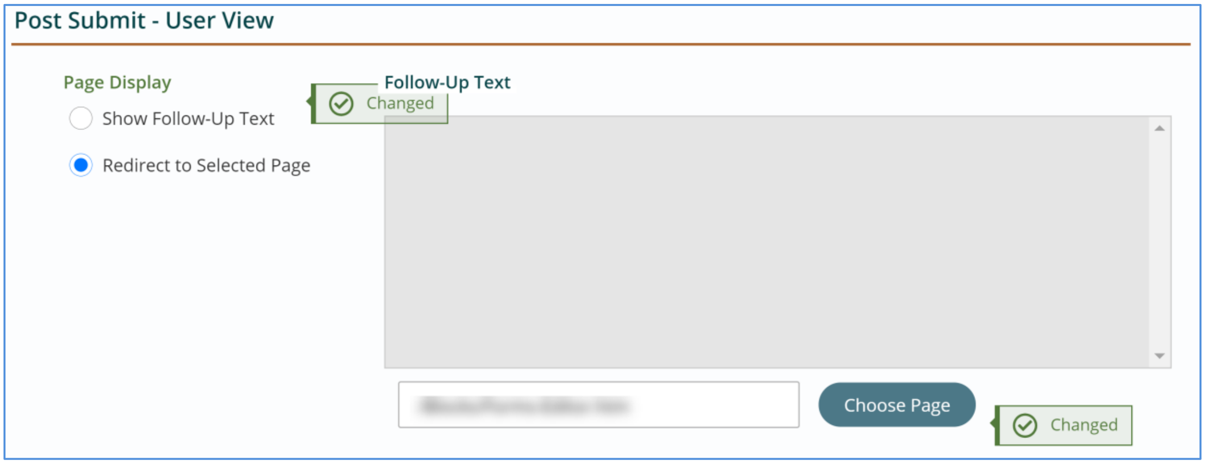
○ Redirect to Selected Page

Follow-Up Text

You can enter text that the user will see after submitting the form.

1. Click the Edit link.
2. Use the WYSIWYG text editor to enter the text that will appear when the visitor submits the form.
3. Click Done.

Redirect to another page following form submission (Redirect to Page)



Post Submit - User View

Page Display

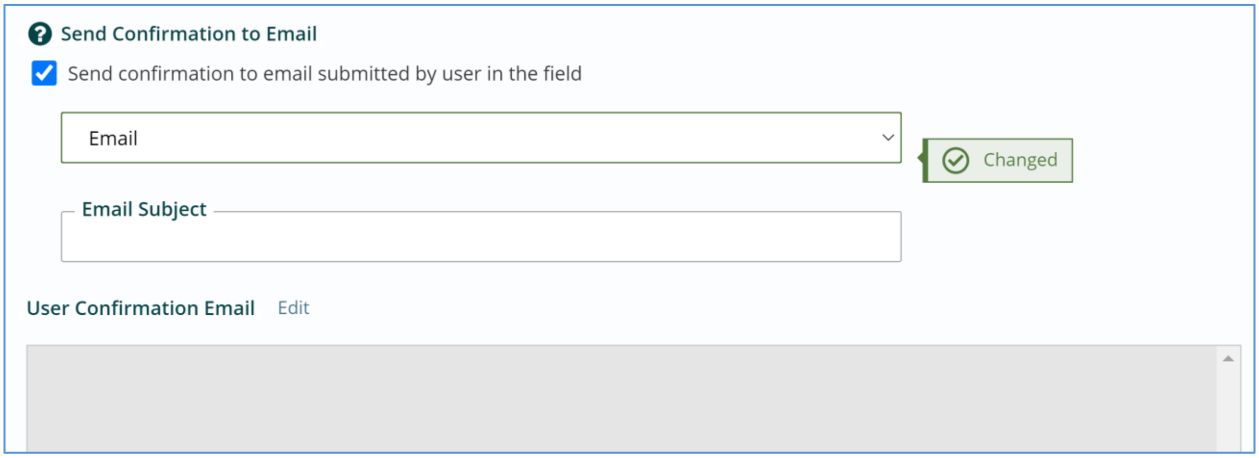
○ Show Follow-Up Text

● Redirect to Selected Page

Follow-Up Text

1. Select the Redirect to Selected Page radio button.
2. Click the Choose Page Button to browse to find the page in your Navigation Tree.
3. Publish.

Send Confirmation Email



Send Confirmation to Email

☑ Send confirmation to email submitted by user in the field

Email

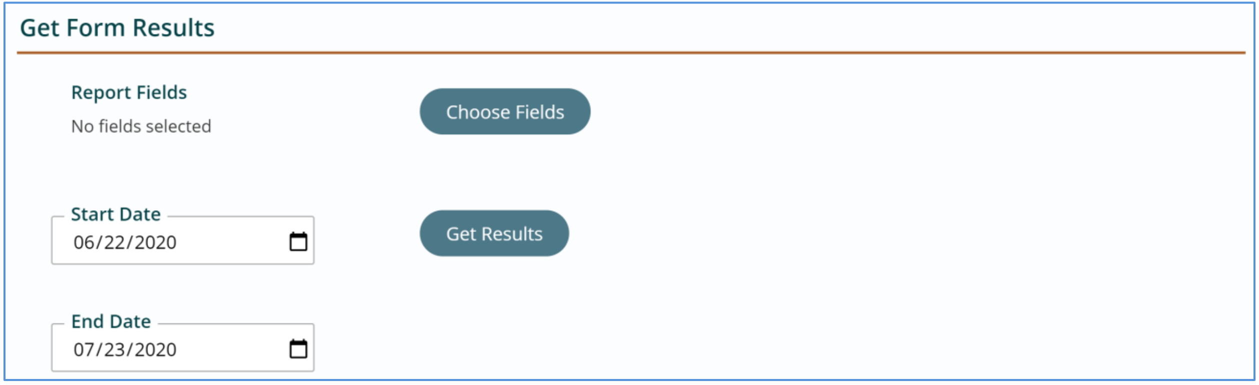
Email Subject

User Confirmation Email Edit

1. Click the Send User Email checkbox.
2. From the drop-down list, select the email field. **Note: An Email Text Field with Verification of the format is required to be built in the form to show up in this list. This field will be used on the form to collect email addresses**.
3. Enter a subject in the Email Subject field.
4. Click the Edit link for User Confirmation Email. The link will only display when checkbox is checked.
5. Build the content for the User Confirmation email in the content section using the WYSIWYG options.
6. Click done.

View Form Submissions

The last portion of the Forms Editor block allows you to view the results of submissions. By default, it shows the last 30 days.



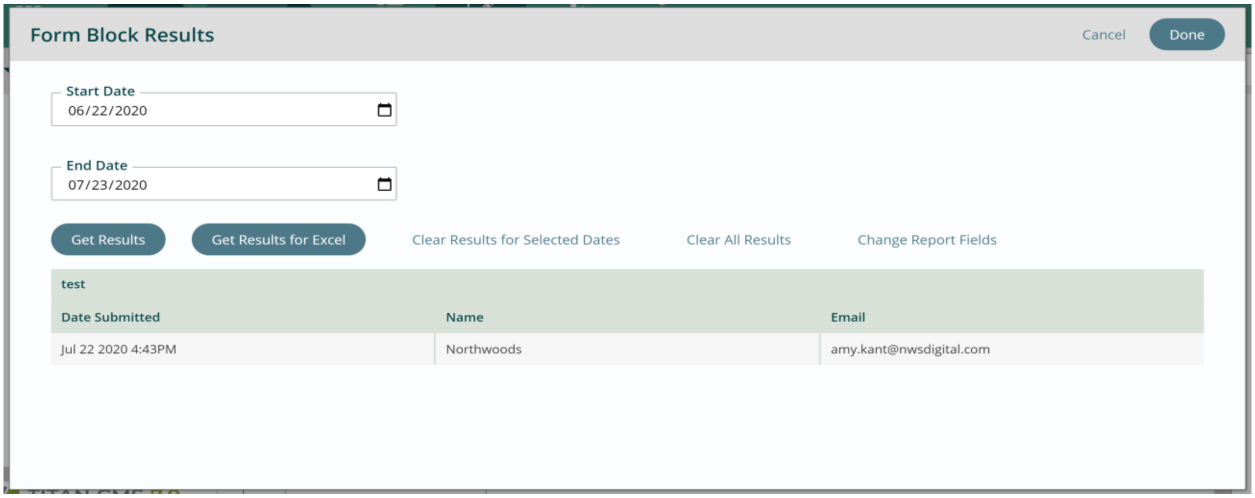
Get Form Results

Report Fields
No fields selected

Start Date: 06/22/2020

End Date: 07/23/2020

The Get Results functionality is also available from your Titan CMS Workstation dashboard. You can configure your dashboard to include this widget. As an alternative, you can open the Forms Editor block on the page and scroll to the very bottom where you will find the Get Results section of the block.



Form Block Results

Start Date: 06/22/2020

End Date: 07/23/2020

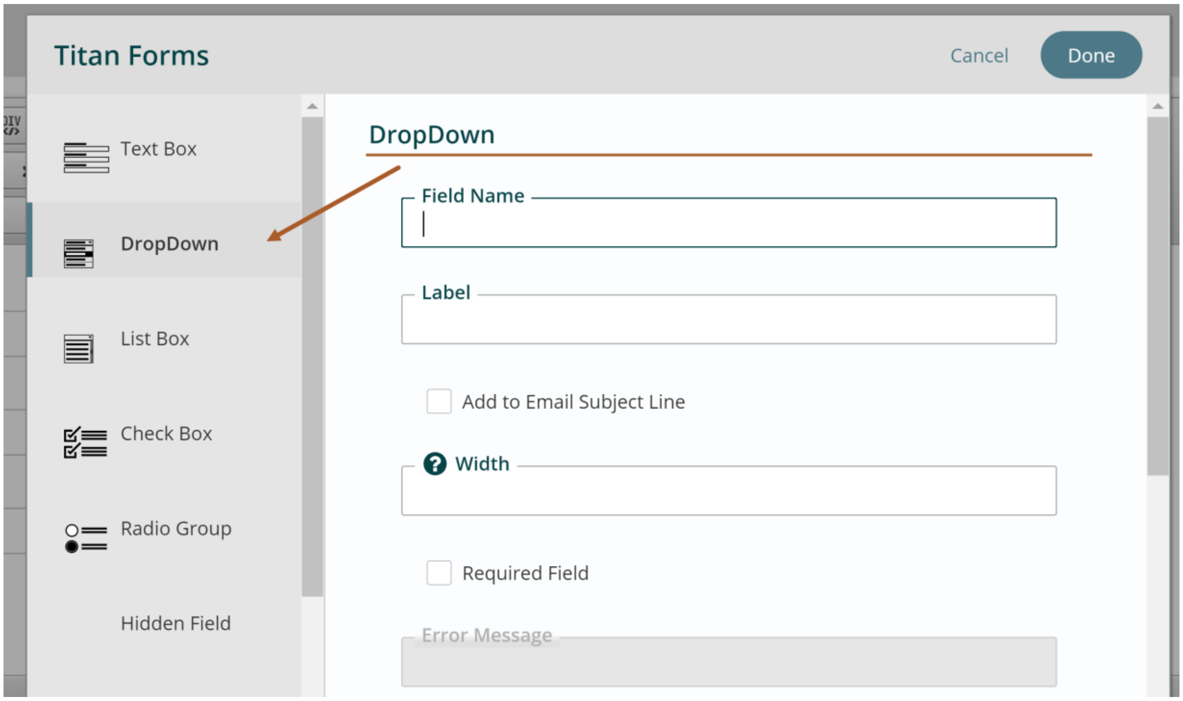
test

Date Submitted	Name	Email
Jul 22 2020 4:43PM	Northwoods	amy.kant@nwsdigital.com

- Use the date fields to filter your results.
- Get Results is clicked after you have changed your Start or End Date fields.
- Get Results for Excel will open the data in an Excel spreadsheet.
- Clear Results will clear everything out of the database (not just your filtered results).
- Change Report Fields will allow you to select those fields you wish to display. You can change these at any time – you won't lose your data if you delete fields.

Form Field Options

Drop Down List Field



Titan Forms

Cancel Done

Text Box
DropDown
List Box
Check Box
Radio Group
Hidden Field

DropDown

Field Name

Label

□ Add to Email Subject Line

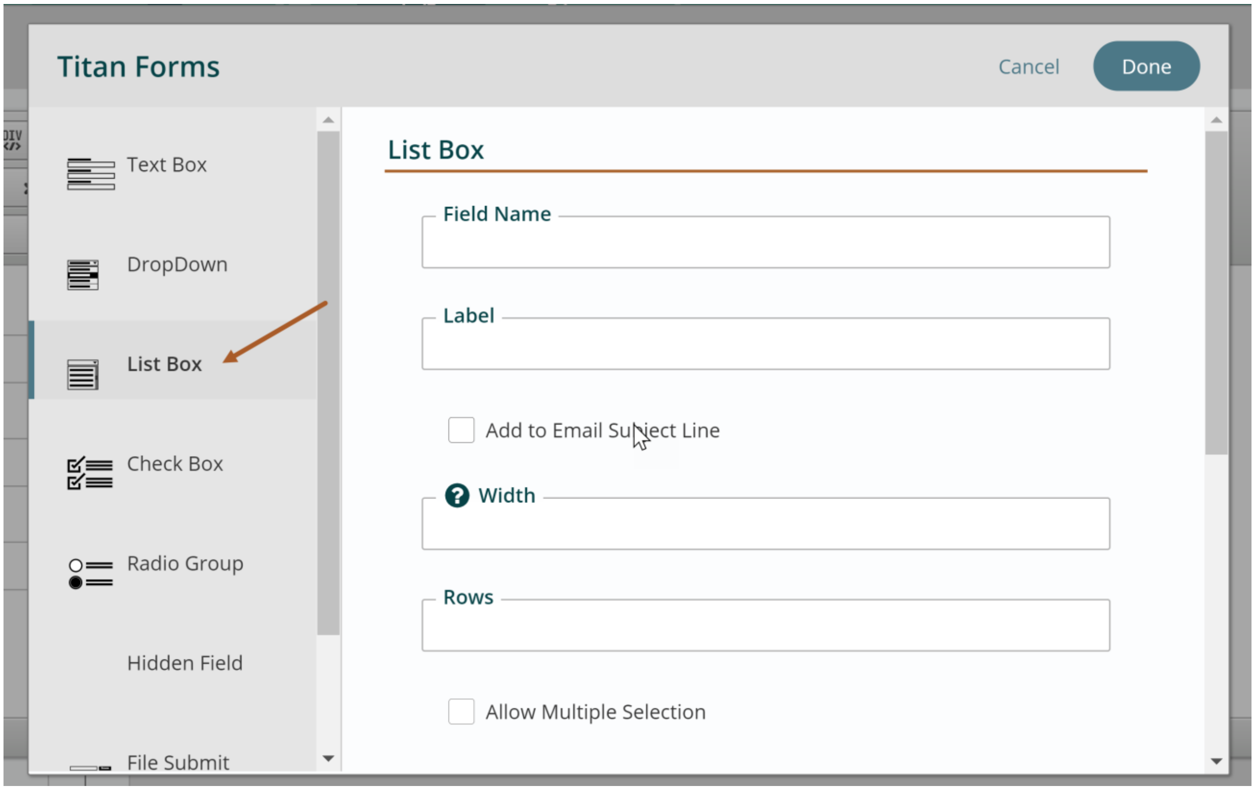
Width

□ Required Field

Error Message

1. Click the Titan Form Icon in the WYSIWYG tool. The Text Box List template will display.
2. Select Drop Down List from the left navigation list.
3. Enter the Drop Down List information.
4. Select Done to add Drop Down List to form. The Form's Content screen will display.
5. Click Save or Publish from the App Ribbon.

List Box Field


Titan Forms

Cancel

Done

Text Box

DropDown

List Box

Check Box

Radio Group

Hidden Field

File Submit

List Box

Field Name

Label

□ Add to Email Subject Line

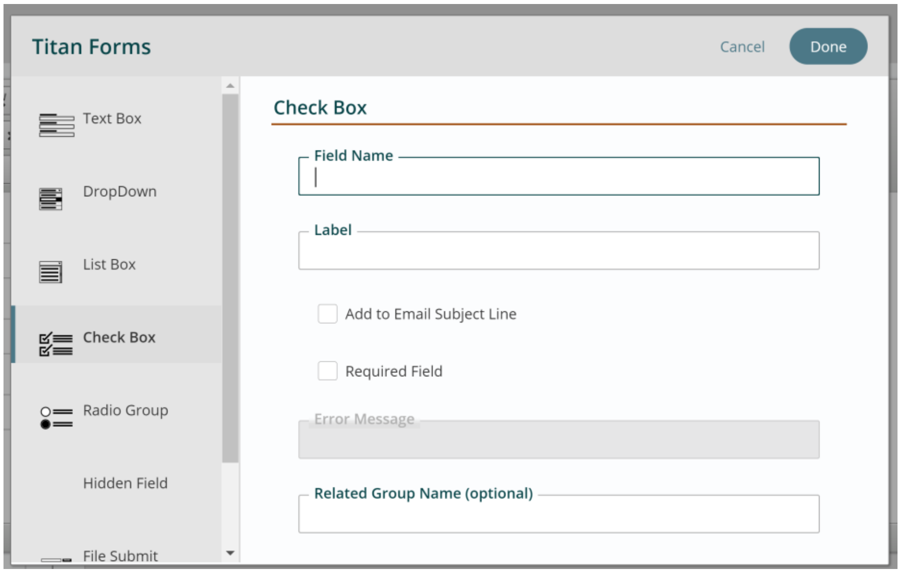
Width

Rows

□ Allow Multiple Selection

1. Click the Titan Forms Icon in the WYSIWYG tool. The Text Box template will display.
2. Select the List Box from the left navigation list.
3. Enter the List Box information.
4. Select Done to add List Box to form or Cancel. The Form's Content screen will display.
5. Click Save or Publish from the App Ribbon.

Check Box Field



Titan Forms

Cancel Done

Text Box
DropDown
List Box
Check Box
Radio Group
Hidden Field
File Submit

Check Box

Field Name

Label

□ Add to Email Subject Line

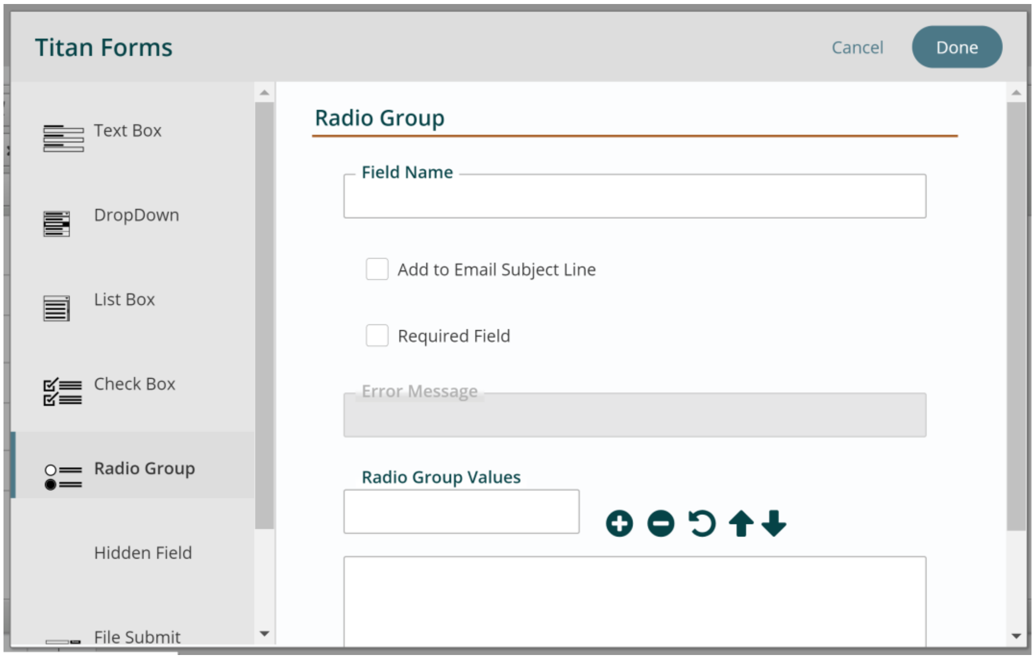
□ Required Field

Error Message

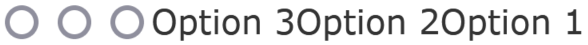
Related Group Name (optional)

1. Click the Titan Forms Icon in the WYSIWYG tool. The Text Box template will display.
2. Select Check Box from the left navigation list.
3. Enter the Check Box information.
4. Select Done to add Check Box to form or Cancel. The Form's Content screen will display.
5. Click Save or Publish from the App Ribbon.

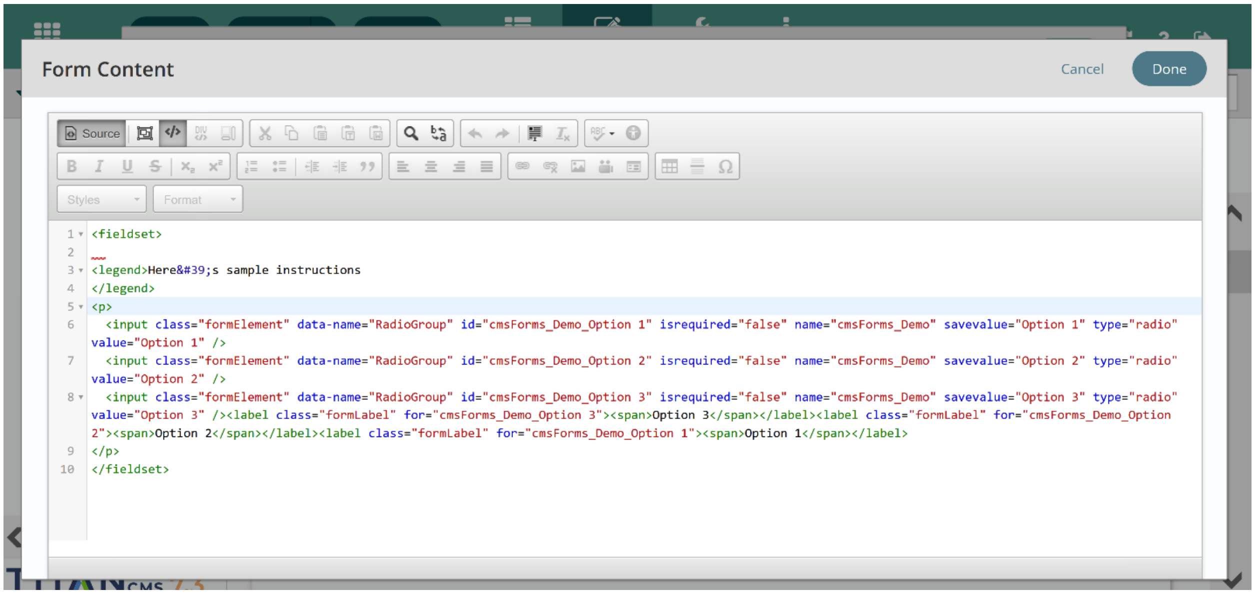
Radio Group Field



1. Click the Titan Forms Icon in the WYSIWYG tool. The Text Box template will display.
2. Click on Radio Group in the left navigation list.
3. Enter the field name and select the checkboxes if relevant.
4. Enter the options into the Radio Group Values field, hitting + after each option. If you would like to rearrange the list, click on an option and hit the up or down arrow. Hit the – button to delete an option.
5. Select Done. The Form's content screen will display, and the buttons will be out of order. Titan CMS is continuously being improved, and this is resolved in Titan CMS 7.4.

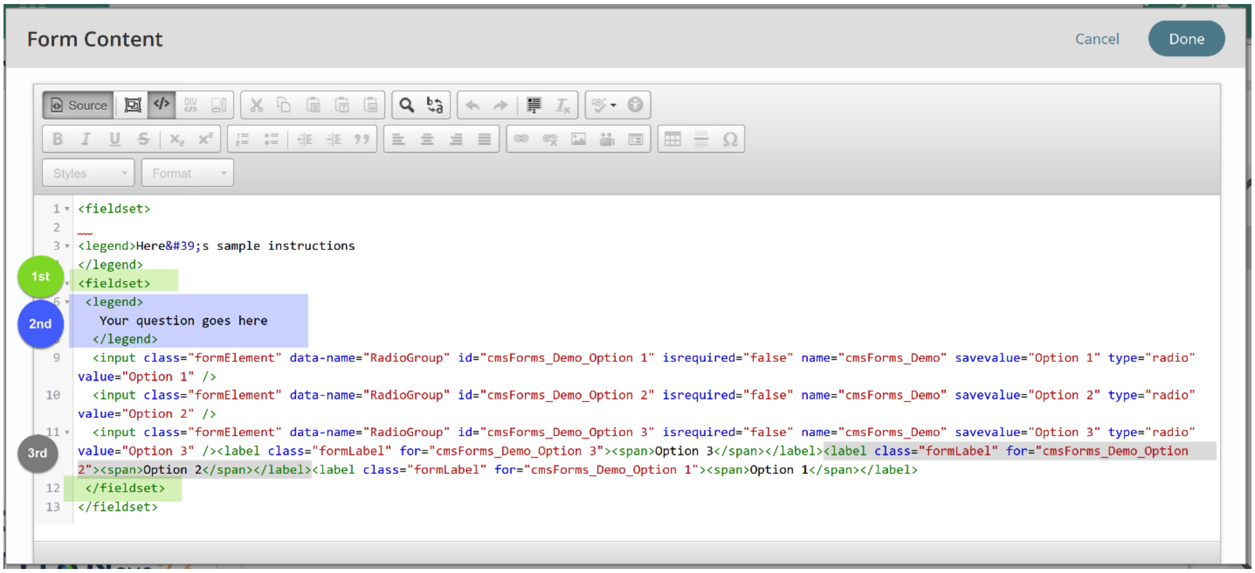


6. Type out the question associated with the radio group above the buttons in the WYSIWYG editor.
7. For the moment, Radio Groups in 7.3 require you to go in to the source code to edit. It will initially look like this:

8. 

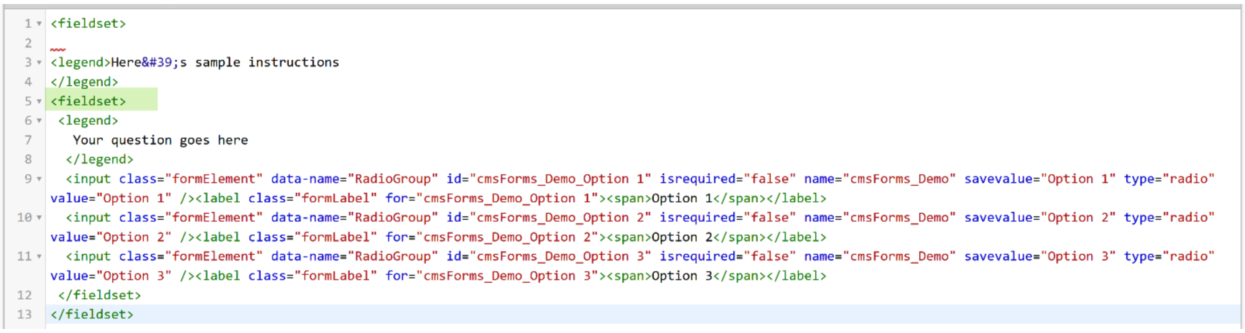
```
<fieldset>

<legend>Here&#39;s sample instructions
</legend>
<p>
  <input class="formElement" data-name="RadioGroup" id="cmsForms_Demo_Option 1" isrequired="false" name="cmsForms_Demo" savevalue="Option 1" type="radio" value="Option 1" />
  <input class="formElement" data-name="RadioGroup" id="cmsForms_Demo_Option 2" isrequired="false" name="cmsForms_Demo" savevalue="Option 2" type="radio" value="Option 2" />
  <input class="formElement" data-name="RadioGroup" id="cmsForms_Demo_Option 3" isrequired="false" name="cmsForms_Demo" savevalue="Option 3" type="radio" value="Option 3" /><label class="formLabel" for="cmsForms_Demo_Option 3"><span>Option 3</span></label><label class="formLabel" for="cmsForms_Demo_Option 2"><span>Option 2</span></label><label class="formLabel" for="cmsForms_Demo_Option 1"><span>Option 1</span></label>
</p>
</fieldset>
```

9. The first thing you will do in the source code is to mark the radio group as a fieldset by adding <fieldset></fieldset> around the entirety of the group.



```
<fieldset>

<legend>Here&#39;s sample instructions
</legend>
<fieldset>
 <legend>
   Your question goes here
  </legend>
  <input class="formElement" data-name="RadioGroup" id="cmsForms_Demo_Option 1" isrequired="false" name="cmsForms_Demo" savevalue="Option 1" type="radio" value="Option 1" />
  <input class="formElement" data-name="RadioGroup" id="cmsForms_Demo_Option 2" isrequired="false" name="cmsForms_Demo" savevalue="Option 2" type="radio" value="Option 2" />
  <input class="formElement" data-name="RadioGroup" id="cmsForms_Demo_Option 3" isrequired="false" name="cmsForms_Demo" savevalue="Option 3" type="radio" value="Option 3" /><label class="formLabel" for="cmsForms_Demo_Option 3"><span>Option 3</span></label><label class="formLabel" for="cmsForms_Demo_Option 2"><span>Option 2</span></label><label class="formLabel" for="cmsForms_Demo_Option 1"><span>Option 1</span></label>
 </fieldset>
</fieldset>
```

10. Second, around the question, change the element from a <p> tag to a <legend> tag.
11. Third, rearrange the labels for the buttons so that they are in the correct order.

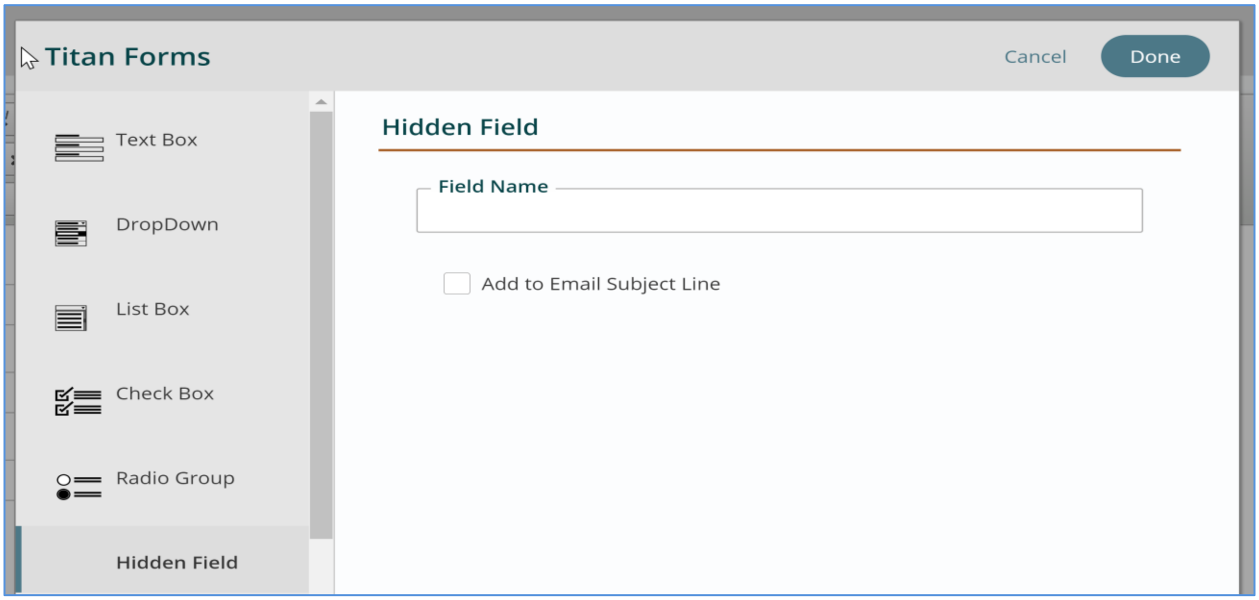


```
<fieldset>

<legend>Here&#39;s sample instructions
</legend>
<fieldset>
 <legend>
   Your question goes here
  </legend>
  <input class="formElement" data-name="RadioGroup" id="cmsForms_Demo_Option 1" isrequired="false" name="cmsForms_Demo" savevalue="Option 1" type="radio" value="Option 1" /><label class="formLabel" for="cmsForms_Demo_Option 1"><span>Option 1</span></label>
  <input class="formElement" data-name="RadioGroup" id="cmsForms_Demo_Option 2" isrequired="false" name="cmsForms_Demo" savevalue="Option 2" type="radio" value="Option 2" /><label class="formLabel" for="cmsForms_Demo_Option 2"><span>Option 2</span></label>
  <input class="formElement" data-name="RadioGroup" id="cmsForms_Demo_Option 3" isrequired="false" name="cmsForms_Demo" savevalue="Option 3" type="radio" value="Option 3" /><label class="formLabel" for="cmsForms_Demo_Option 3"><span>Option 3</span></label>
 </fieldset>
</fieldset>
```

12. Click Done in the Form content, then Save or Publish from the App Ribbon.

Hidden Field

With the use of a special script, hidden fields of information may be passed to a 3rd party website (user doesn't see). The Hidden Field uses the OnLoad Function field in the Forms Editor block with a special script created by developer. Contact your Titan Administrator for assistance.



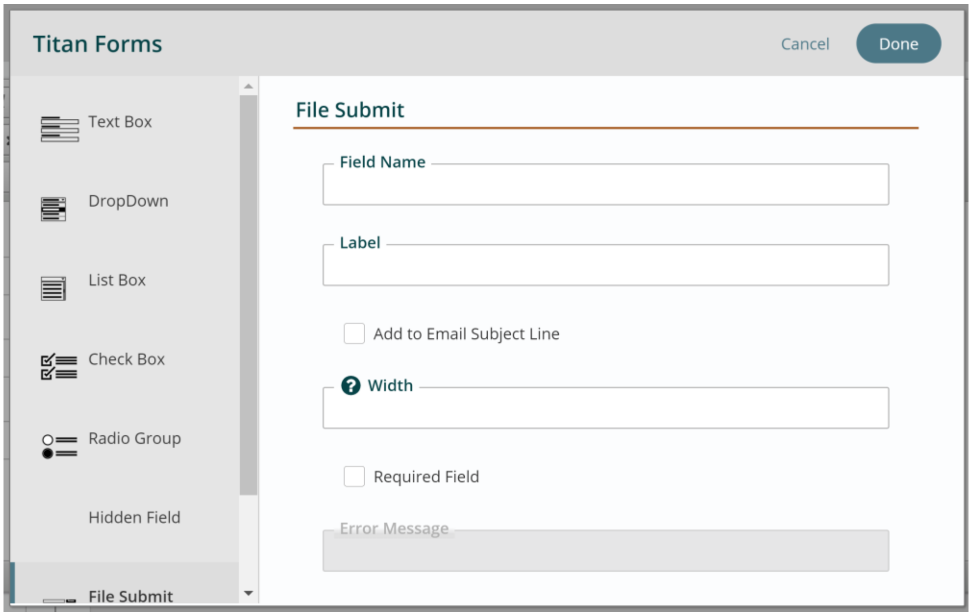
Hidden Field

Field Name

□ Add to Email Subject Line

File Submit Field

The File Submit Titan Form field allows a user to upload a file through your form on your website. The Form Submission Action must be POST to Titan, this allows you to select where to store those uploaded files.



1. Place your cursor in form where you wish to add the File Submit field.
2. Click the Titan Forms Icon in the WYSIWYG tool.
3. Select the File Submit option. You may need to scroll to find it.
4. Enter the File Submit information.
5. Select Done to add File Submit field to form. The form's content screen will display.
6. Click Save or Publish from the App Ribbon.

Field Labels

LABEL	DESCRIPTION/USE
Field Name	This is the name that is used in the database and on the spreadsheet report. Use a name that will represent the question, title, or label that you have added in your content.
Add to Email Subject Line	If this is checked, the field name(s) for any selected fields built on the form will be added to email subject line.
CSS Class	The default value is automatically selected. You can change value if other CSS classes are listed in drop down.
Width	This is the setting for your field’s width.
Rows	This is the setting to allow for 1 or more rows within your text field.

LABEL	DESCRIPTION/USE
Allow Multiple Selection	When checked, this setting will allow someone to select more than 1 value in your box.
Required Field	Check this box if your field will be required on your form.
Error Message	When the user doesn’t fill out the field, this message will advise that this field is required.
Display Value	This is the value that will display on the website. This entry may include spaces.
Database Value	The value that will be assigned to the Display Value entered above and will be stored in the database. Spaces are not allowed.
Add As New	This is the function button to add your entered Display Value in the list. When using a Drop Down or List Box, make your first entry "no selection" or "none" to force user to select from list.
Update Selection	This is the function button to change your selected Display Value and Database Value. When you click on the Display Value in the list, it will be highlighted, and the function buttons Update Selection and Delete Selection will be enabled. Make your change and click Update Selection.
Delete Selection	This is the function button to delete your selected Display Value and Database Value. When you click on the Display Value in list, it will be highlighted, and the function buttons Update Selection and Delete Selection will become enabled. Click on Delete Selection.
Move Up	This is the function link to move your Display Value up in the list. When you click on the Display Value in the list, it will be highlighted. Click on the link for Move Up and Display Value will move one place up. Continue clicking Move Up until it is in the desired location.
Move Down	This is the function link to move your Display Value down in the list. When you click on the Display Value in the list, it will be highlighted. Click on link for Move Down and Display Value will move one place down. Continue clicking Move Down until it is in the desired location.