

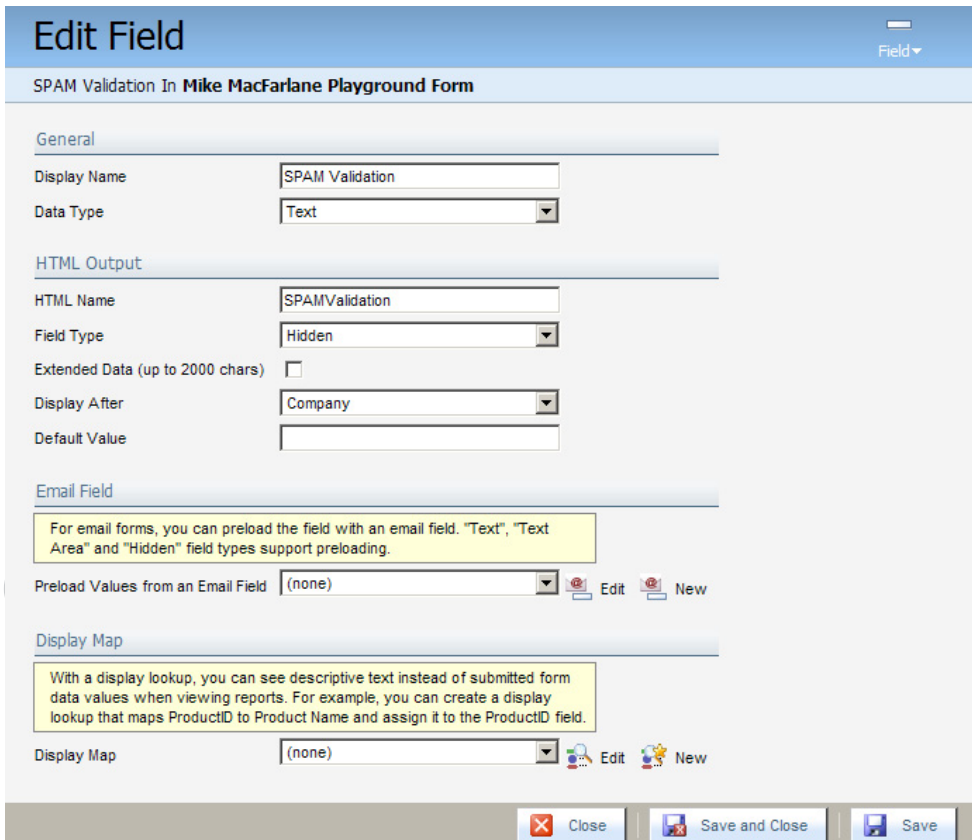
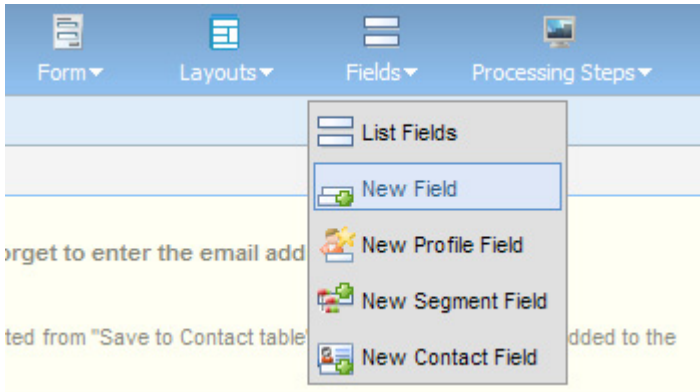
Protecting Your Forms Against Spam



This document provides instructions for a few ways to prevent spam data submissions to your web forms using standard Eloqua® functionality. None of the solutions presented below are a “silver bullet” since focused and dedicated efforts to crack particular forms will often be indefensible, but they are easy practices to adopt when publishing your forms so that you keep spam submissions to a minimum.

Using Hidden Fields to Prevent Spam Submissions

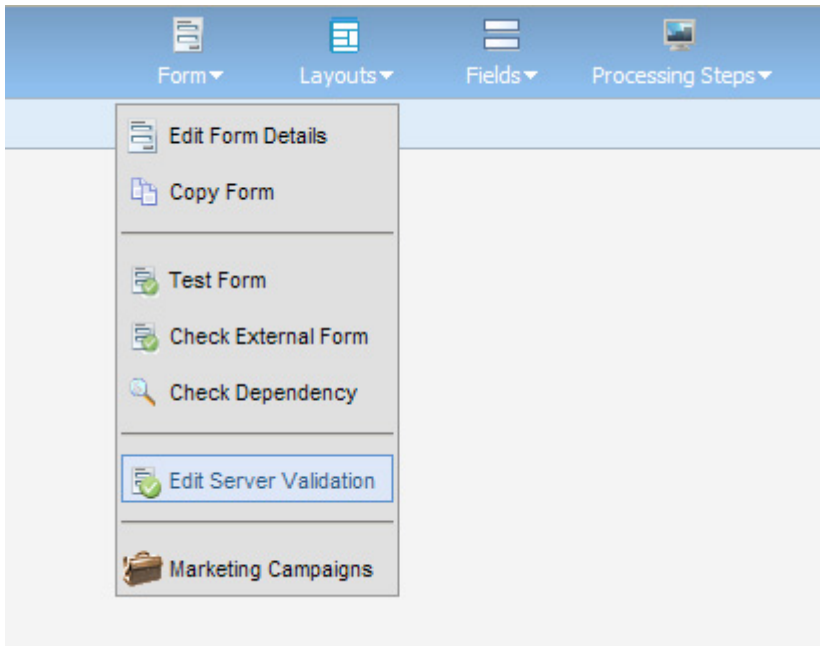
1. Add a new **Hidden** field to your form (in this example we will give the field the name, “SPAM Validation”).



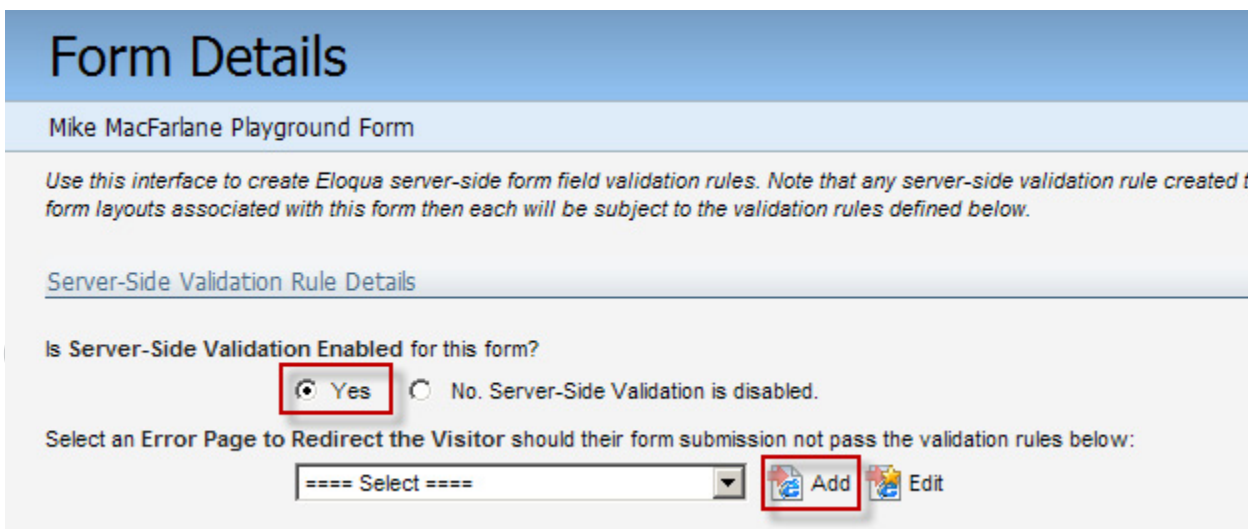
Note: If you are hosting this form, then you will also need to add the following HTML to your form landing page in the appropriate area of the code:

```
<input type="hidden" name="SpamValidation" value="">
```

- 2. Create a **Server Validation** rule that will check to make sure that a value has not been submitted into your hidden field - in the event that a value IS submitted into your hidden field, the system will reject the submission as it will know that it wasn't a real person filling in the form.





Note: In the case where your validation rule is not met, there will be a page presented to the bot/visitor informing them of the error - you will need to define this page as well.



3. Configure and define your validation rule. You will select the field for which you want to have a rule (in this case, "SPAM Validation") and then define the logic of that rule. You will want a **text** rule and you will want that rule to essentially state, "*don't let any submissions come in where the value submitted in the hidden field, "SPAM Validation" is longer than 1 character.*"


Server-Side Validation Rule Details

Is Server-Side Validation Enabled for this form?
 Yes No. Server-Side Validation is disabled.

Select an Error Page to Redirect the Visitor should their form submission not pass the validation rules below:
==== Select ====  Add  Edit

Server-Side Validation Rules

You have yet to define any Server-Side Validation rules.

Create New Server-Side Validation Rule: ==== Select ====  New

- Email Address
- First Name
- Last Name
- Company
- SPAM Validation**
- ==== Select ====

New Field Validation - SPAM Validation

Form Field

Field Name SPAM Validation

Form Field Validation

Validation Type Text

(Continued on next page)

New Field Validation - SPAM Validation

Form Field

Field Name SPAM Validation


Form Field Validation

Validation Type

Error Messages On Failure

Server Message

An Invalid or Empty Value was provided for elqUserFormFieldDisplayName. [elqUserFormFieldValueSubmitted]

Field Value Submitted  Insert Custom Field

If you intend to create Client-Side Validation code from this rule please indicate below the error message to appear on failure. To pre-populate the message with the current field name use the token [%field%]. (ie. Form field [%field%] is invalid.)

JavaScript Message

Form Field Validation Parameters

Parameter Name	Enabled	Source	Parameter Value
Set to this Default Value If Empty	<input type="checkbox"/>	Constant	<input type="text"/>
Minimum (Length)	<input type="checkbox"/>	Constant	<input type="text"/>
Maximum (Length)	<input checked="" type="checkbox"/>	Constant	1

4. Save and Close - you're done!

Using a Question and Answer Field, or CAPTCHA, to Prevent Spam Submissions

Note: CAPTCHA stands for, "Completely Automated Public Turing test to tell Computers and Humans Apart"

In this solution, you will be creating a new field that will ask the visitor a question that has only one possible answer (or a few versions of one answer) and then set up validation to ensure that the value submitted was the one(s) identified as valid. For this example, we will use the question, "What is 2+2?"

1. Add a new standard field to your form (in this example we will again give it the name, "SPAM Validation"). Standard text field, same defaults as above.
2. Create a **Server Validation** rule that compares the data submitted in your Spam Validation field to a **Data Value Set** of valid "answer(s)" to the question posed for this field. Given that the question is, "What is 2+2?", we will input the valid values of, "4" and "four" (validation is NOT case-sensitive) as valid answers.

Server-Side Validation Rules

You have yet to define any Server-Side Validation rules.

Create New Server-Side Validation Rule: Spam Validation  New

New Field Validation - Spam Validation

Form Field


Field Name Spam Validation

Form Field Validation

Validation Type Field value in Data Set

Error Messages On Failure



Server Message An Invalid or Empty Value was provided for elqUserFormFieldDisplayName. [elqUserFormFieldValueSubmitted]

Field Value Submitted  Insert Custom Field

If you intend to create Client-Side Validation code from this rule please indicate below the error message to appear on failure. To pre-populate the message with the current field name use the token [%field%]. (ie. Form field [%field%] is invalid.)

JavaScript Message

Form Field Validation Parameters

Parameter Name	Enabled	Source	Parameter Value
Set to this Default Value If Not in Set	<input type="checkbox"/>	Constant	
Data Value Set to Verify	<input checked="" type="checkbox"/>	Constant	city  

Value Set

Valid Answers to 2+2

OVERVIEW - Add one or more values. If you need to add more values, click the plus icon. The first 10 values entered are displayed.

Add Values

Value

Value

Value

Form Field Validation Parameters			
Parameter Name	Enabled	Source	Parameter Value
Set to this Default Value If Not in Set	<input type="checkbox"/>	Constant	<input type="text"/>
Data Value Set to Verify	<input checked="" type="checkbox"/>	Constant	Valid Answers to 2+2 Edit New

- For the visible field name, make sure it reads “What is 2+2?” (and you may want to add a little text explaining why you’re asking them to answer that question).

Note: Again, if you are hosting this form, then you will also need to do all of the above in Eloqua® AS WELL AS edit the HTML of the ‘front-end’ form.

You’re done!

If you have any questions or comments, please contact Eloqua® support.

