

Simfatic Forms

User's Guide



Simfatic Forms

Version 4.0

Read the online version of this document here:

<http://www.simfatic.com/forms/help/v40/>

<http://www.simfatic.com>

Contents

Chapter 1	Introduction	12
	How does an HTML Form work?	14
Chapter 2	Getting Started	18
	Create a simple form and get it online	19
	Step 1: Let's Plan a Simple Form	20
	Step 2: Add Input Elements	22
	Step 3: Add Validations	26
	Step 4: Select Form Processing Options	29
	Step 5: Edit Email Template	31
	Step 6: Edit 'Thank you' Page	35
	Step 7: Upload the Form	37
	Enhancements: adding more features to the form	40
	Save the form submission data to a database table	41
	Send an auto-response to your visitor	43
Chapter 3	How To ...	46
	How to save form submissions to a file	47
	How to send an auto response	51
	How to make changes to an already installed form	53
	How to display elements conditionally	54
	How to email based on a selection	56
	How to create a simple calculation field	60
	How to create an online calculator	65
	How to load list box items from a file	71
	How to create a cascaded list	74
	How to change the form page size	83
	How to add a conditional validation	84

Contents

How to add your own validations	86
How to upload the form manually	88
How to troubleshoot form errors	95
How to install the form on a subdomain	97
How to add custom functionality to your form	99
How to edit the validation error messages	101
How to pass a value to the form	102
How to accept payments through Paypal	105
How to handle JavaScript events in the form	107
How to customize the Look & Feel of the form	111
How to redirect form submission values to another script	113
How to make a Terms & Conditions box	114
How to seamlessly integrate with your website	115
How to Integrate the form	116
Dreamweaver	118
Expression Web	119
Joomla	120
Drupal	121
WordPress	122
How to make a popup form	123
How to view form submissions online	127
Chapter 4 Information Reference	132
The Simfatic Forms project file	133
Naming Rules	135
Enter correct 'From' address for proper Email delivery	137
Calculation Field	139
Calculation field : function reference	143
Using the duplicate feature	151
Is Captcha required in your form?	152

Contents

Prefer File uploadbox rather than Native file upload	154
Smart Validations	156
Unique validation requires Database	157
Extension Modules	158
Function Reference	163
Update Configuration Settings	177
Why do I need to upload form to a webserver ?	179
Error when selecting web publish folder	181
Features that requires database support	183
Turn logging off for better performance	184
Scared of write permissions to formdata folder	186
Limitations of 'full form code'	188
Chapter 5 UI Reference	192
The First Page	193
Create form using a template	196
Menu Options	198
Previewing a form	201
Options dialog	202
The General Layout of the Wizard	205
General Form Properties Page	207
Look and feel page	211
'Draw the form' Page	214
Element properties	219
Custom code dialog box	260
Tab order dialog box	263
Page settings dialog box	265
'Add Input Validations' page	266
Validation Types	271
Validation condition dialogbox	283

Contents

Form Processing Options Page	285
Configure Form-mail Page	289
The 'Reply-To' field dialog box	293
'Edit Confirmation Page Template' page	294
'Auto-response' page	296
'Form to File' Page	299
'Save to database' page	301
Setup PayPal Page	303
'Extension Modules' Page	306
'Thank You!' Page	307
Take the code page	309
Take the code Wizard	311
SMTP settings dialog box	314
'Copy and Paste Form Code' dialog box	315
Code Generation Settings dialog box	317
The Add Server Wizard	319
My Servers Dialog Box	323
Generate code page ('client side only' case)	324
Form installation maintenance page	326
Form Admin Page	329
Download log file dialog box	331
Download form submission data file dialog box	333
Form submission data viewer	336
Common Dialog Boxes	338
Address Book	339
Insert Field Table	341
HTML Editor	343
Find Element Dialog	345
Imported Forms	346

Contents

Read form from a web page

347

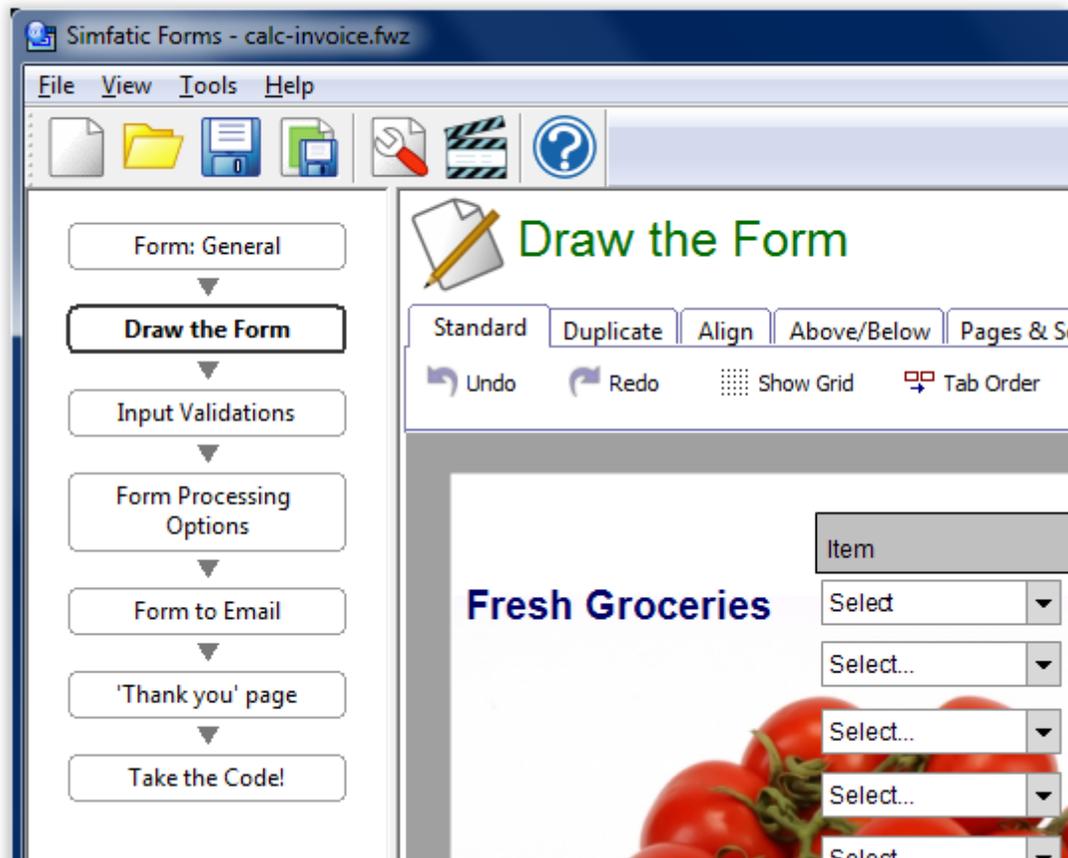
General Options

349

Part



Introduction



Simfatic Forms is a web form builder.

Using Simfatic Forms you can

- Visually design web forms with different types of input elements
- attach validations to the input elements
- get form submissions by email
- get the form submissions to a database table

- send an auto-response to the visitor
- do payment processing through Paypal
- and more!

In addition to creating your online forms, you can upload and maintain your web forms using Simfatic Forms.

This help documentation contains getting started guide and a detailed reference

System Requirements

- Microsoft Windows XP/Vista/7.

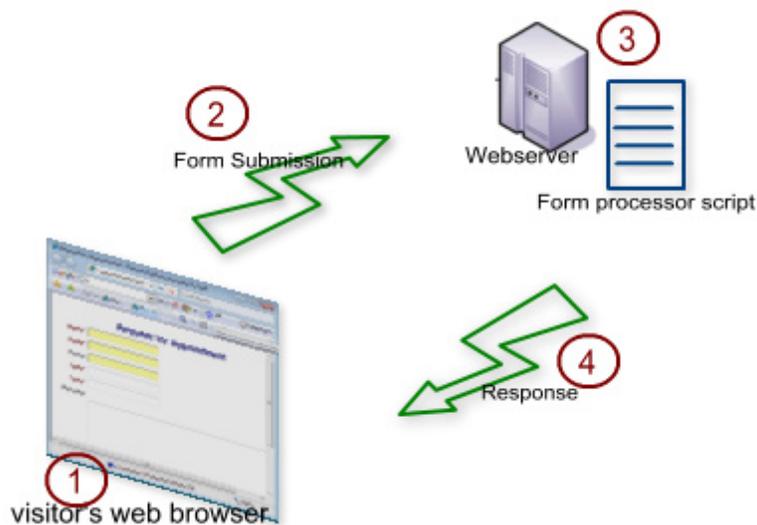
Simfatic Forms is a desktop application that runs on MS Windows systems

- Your web site should support PHP 4.3.0 or above.

Please note that your website could be running under any platform (Windows or Linux). The only requirement is that it should support PHP 4.3.0 and above. PHP(<http://www.php.net>) is a free, open source server-side scripting language.

1.1 How does an HTML Form work?

This is a beginner's tutorial for understanding the working of a web form and the different parts involved.

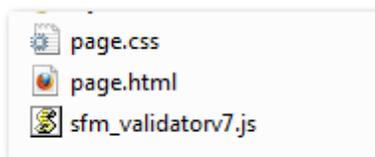


1. Your visitor loads the form page in her web browser.

The browser sends a request to the web server. The web server returns the **HTML** page that contains the form.

The HTML page returned can be

- a HTML file on the server
- an HTML page generated by a script running on the web server (like, a **PHP** script)



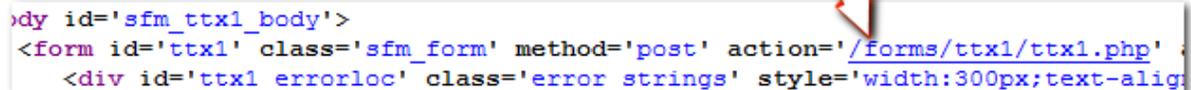
The HTML page can contain references to **Cascaded Style Sheets**, **Javascript** files and images. The web browser downloads them as well. Then the page is displayed.

The Style sheets (also known as **CSS** files) contains the visual aspects (fonts, colors, etc) of the page. The JavaScript files are to control the behavioral aspects (input validations, messages etc).

2. Your visitor fills the form and submits it

Well written forms will have a bit of Javascript code that gets triggered when the form is submitted. The Javascript code checks the form submission values and informs your visitor if there is any error. If there is no error, the form is submitted.

3. The form submission data is sent to the web server

A screenshot of HTML code showing a form tag. A red arrow points to the 'action' attribute. The code is:

```
body id='sfm_ttx1_body'>  
<form id='ttx1' class='sfm_form' method='post' action='/forms/ttx1/ttx1.php'  
<div id='ttx1_errorloc' class='error_strings' style='width:300px;text-align:
```

Once the visitor has submitted the form, the form data is sent to the web server. In the form, the author of the form has to mention an 'action' URL that tells the browser where to send the form submission data. The 'action' usually points to the URL of a script that knows what to do with the data

4. The web server processes the request

The web server passes the form submission data to the **form processor script** (mentioned by the 'action'). The form processor script is written in languages like PHP, ASP, or Perl. The form processor script can send the form data by email, save the data to a database or file.

5. A response is sent back to the browser

The form processor script sends a response indicating the success or failure of the

form processing operations back to the server. The response could be to re-direct to another web page.

What does Simfatic Forms do?

Simfatic Forms can generate the code for all the above! The HTML form page, Javascript validations, CSS and the form processor script (in PHP). You just have to design the form. Simfatic Forms generates the code and uploads it to **your webserver**.

The whole form works on your web server.

Part



Getting Started

Using Simfatic Forms is extremely easy. You are guided step by step by the wizard, collecting your requirements. Then on press of a button, you can generate all the code and install on your web site.

This 'Getting started' guide provides a quick introduction to Simfatic Forms.

2.1 Create a simple form and get it online

This guide takes you through a quick tutorial on creating and installing a simple form.

Next: Let's Plan a Simple Form

Step 1: Let's Plan a Simple Form

To start, we will plan a simple feedback form for your web site. Let's suppose we have the following requirements:

- The form should have fields for name, email and comments/feedback.
- There should be validations to check the validity and the size of the input.
- The submitted form data should be sent to an email address (say, myforms@mycompany.com)

'Create a new blank form'

Start Simfatic Forms.

Click on the 'Create New Form' button.

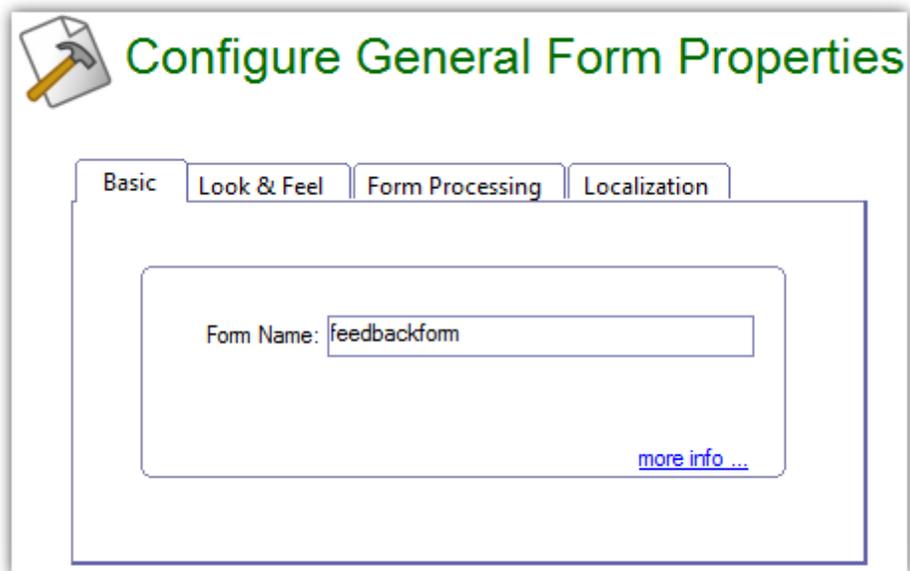


The other options in the first page are for creating form from a template and to read out a form from an existing web page.

When you press on the 'Create a new blank form' button, the wizard screen opens. In this screen, you can see the steps to the left side and the pages in the right side. (See The General Layout of the Wizard)

General form properties

In the general form properties page, you need to provide the basic information of the form. (See General Form Properties Page for more info.)



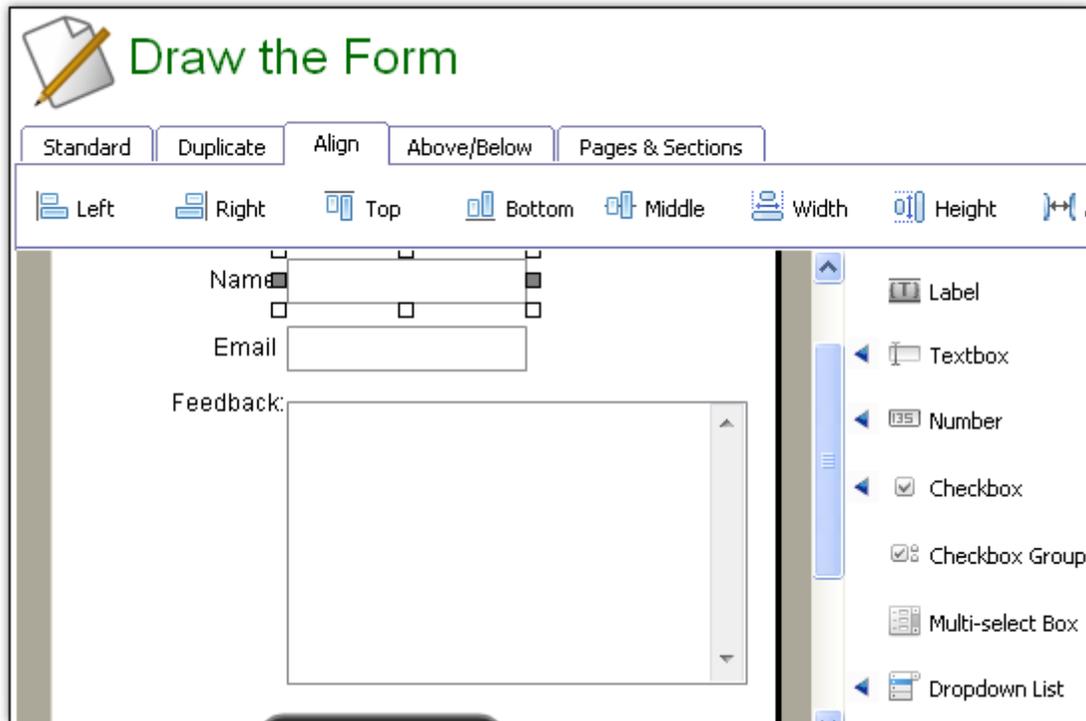
- Provide a name 'feedbackform' in the Form name field.
- Press the 'Next' button

In the next step, we will add the input elements to the form.

Next: Add Input Elements

Step 2: Add Input Elements

The 'Draw the form' page is for adding input elements to your form.

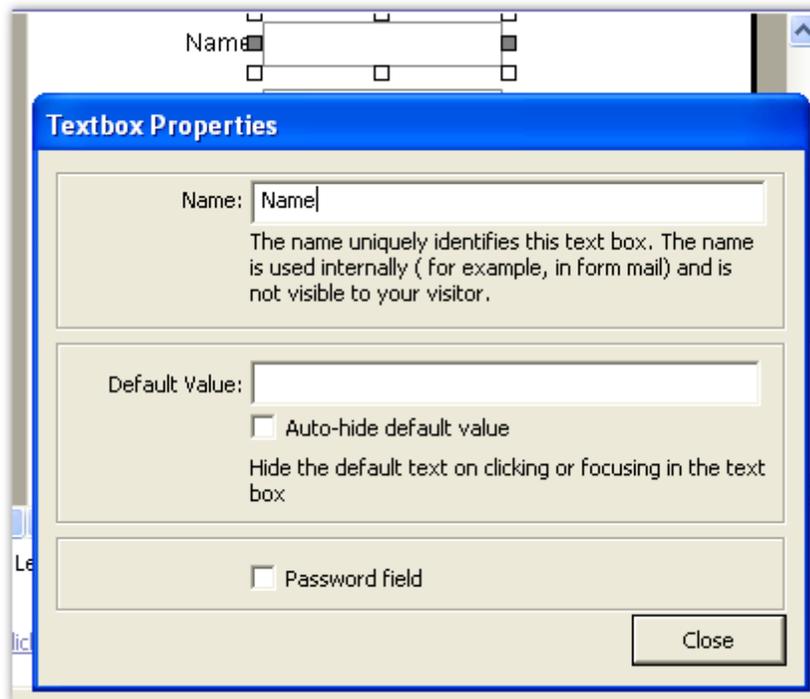


Draw the form page

First, we will add the 'Name' field.

Add the 'Name' field

- Press the 'Textbox' button. A text box is added to the form.
 - Place the text box by dragging it.
 - Double click on the text box to open the properties box for the Textbox.
 - Change the name of the field to 'Name'.



- Press the 'Label' button and place the label next to the 'Name' Text box.
- Open the properties box for the label by double-clicking it.
- Change the label text to 'Name'

Add an 'Email' field

- Add a Textbox for the email field just like we added the 'Name' field. (With name and label as 'Email')

Add a 'Feedback' field

- To add a 'Feedback' field, press the 'Multi-line' Text button. A multi-line text box object is placed on the form.
- Double-click on the multi-line text box and change its name to 'Feedback'

- Add a label for the 'feedback' field and change it's label text to 'feedback'



 For 'Name' and 'Email' we chose 'single line' Textbox where as for feedback we expect multi-line input. Thus we added a multi-line box.

Now we have all the elements required for the form ready.

No; oops! We did not add a 'Submit' button!

Add a submit button

Press the button labelled 'Submit button'.

A submit button is placed on the form.

Let' have a 'Preview'

Click on the preview button to have a quick look at your form.



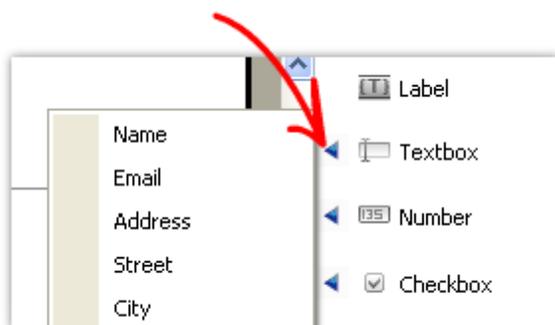
Element Properties

Select an element from the form, right click it and choose 'properties' from the context menu.

Alternatively, select an element from the form, and press Ctrl+Enter to open properties of an element.

Quick sets

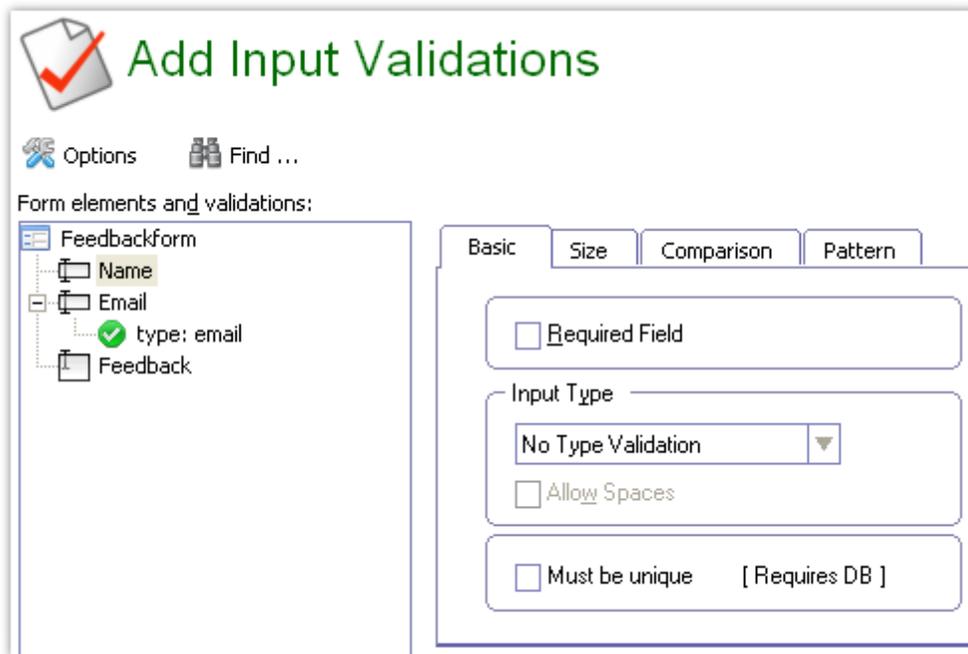
The right arrow buttons next to the input elements are for quickly adding commonly used input elements.



Prev: Let's Plan a Simple Form | Next: Add Validations

Step 3: Add Validations

This page is to add input validations- checks to ensure that the data input is valid. In the 'Add Input Validations' page, the left side tree displays all the input elements. Select an element from the left; the validations for the selected element appears to the right.

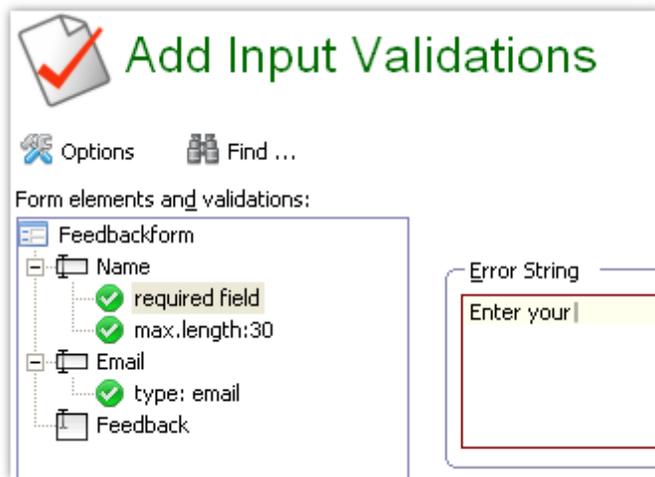


Add validations to the 'Name' field

- Select the 'Name' field from the left side box.
- We will make the 'Name' field mandatory. Select the 'Required Field' check box
- Now select the 'size' tab. Select the 'Max Length' check box and provide a maximum value(40 or 50 ?)

Notice the validation nodes appearing on the left-side tree.

- Select the 'required field' node from the left-side tree.
- On the right side you will see the error message for this validation (the error displayed if this validation fails). You can customize the error message



Validations for the 'Email' field

- Select the 'Email' node from the left side box.
- On the right side, select the 'Required Field' check box, making it a mandatory field
- From the 'Input type' drop down, select 'Email' type

Validations for the 'Feedback' field

We will set a maximum length limit for the feed back field.

- Select the 'Feedback' field from the left side
- Select the 'Maximum Length' check box and enter the Max. limit (say 10240, ie 10 KB?)

Now, we are ready with the input validations.

Save your work

You can, at any point, save your work to a 'Simfatic Forms project' file. (The file extension is fwz)

- Click on the Save button from the toolbar
- The file dialog pops up. Select the file location and the file name.

Press the Next button to move to the next step.

Prev: Add Input Elements | Next: Select Form Processing Options

Step 4: Select Form Processing Options

There are a number of options that you can do with the form submission data. For our feedback form we just want to receive the submitted form data by email.

- Select the 'Send form submission Emails' check box
- Leave all the other options unchecked
- Under 'Troubleshooting', check the 'Debug Mode' option on. This will help in getting detailed errors if the form processing does not work well.
- It will also be good to provide your email address in 'Email address to receive errors' field



Select Form Processing Options

Choose what to do with the form submission data.

 Show a 'confirm the submission' page
(allows your visitor to verify/take a print before submission)

Send emails _____

 Send form submission Emails

Send an auto-response to the visitor who submitted the form

Save submissions _____

 Save the form submission data in a database table

▼ **Troubleshooting**

Enable Logging
All activities in the form are logged to a text file.
You can download the log from the Form Maintenance page

Debug Mode (Display errors on the page itself)

Email address to receive errors:

The script sends an email in case there is an error

Press the Next button to move to the next step.

Prev: Add Validations | Next: Edit Email Template

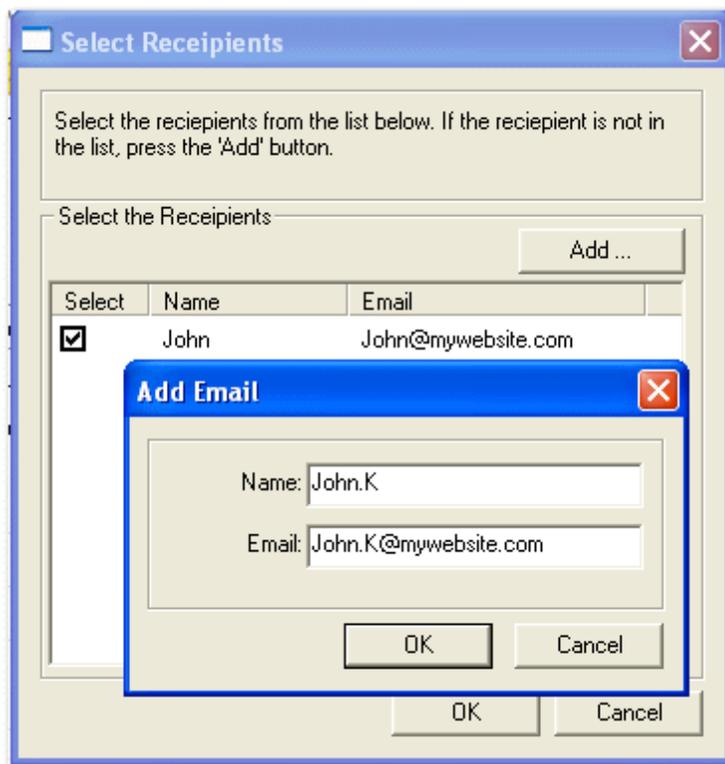
Step 5: Edit Email Template

When someone submits your feedback form, the form data is composed in to an email and is sent to the configured email addresses.

The Configure Form-mail page lets you configure the email sent when the form is submitted.

Configure the Recipients

- Press the 'Select...' button next to the 'Recipients' field
- A 'Select Recipients' box opens Press the 'Add' button to add your name and email address to the list. Then choose the address from the list and press OK.



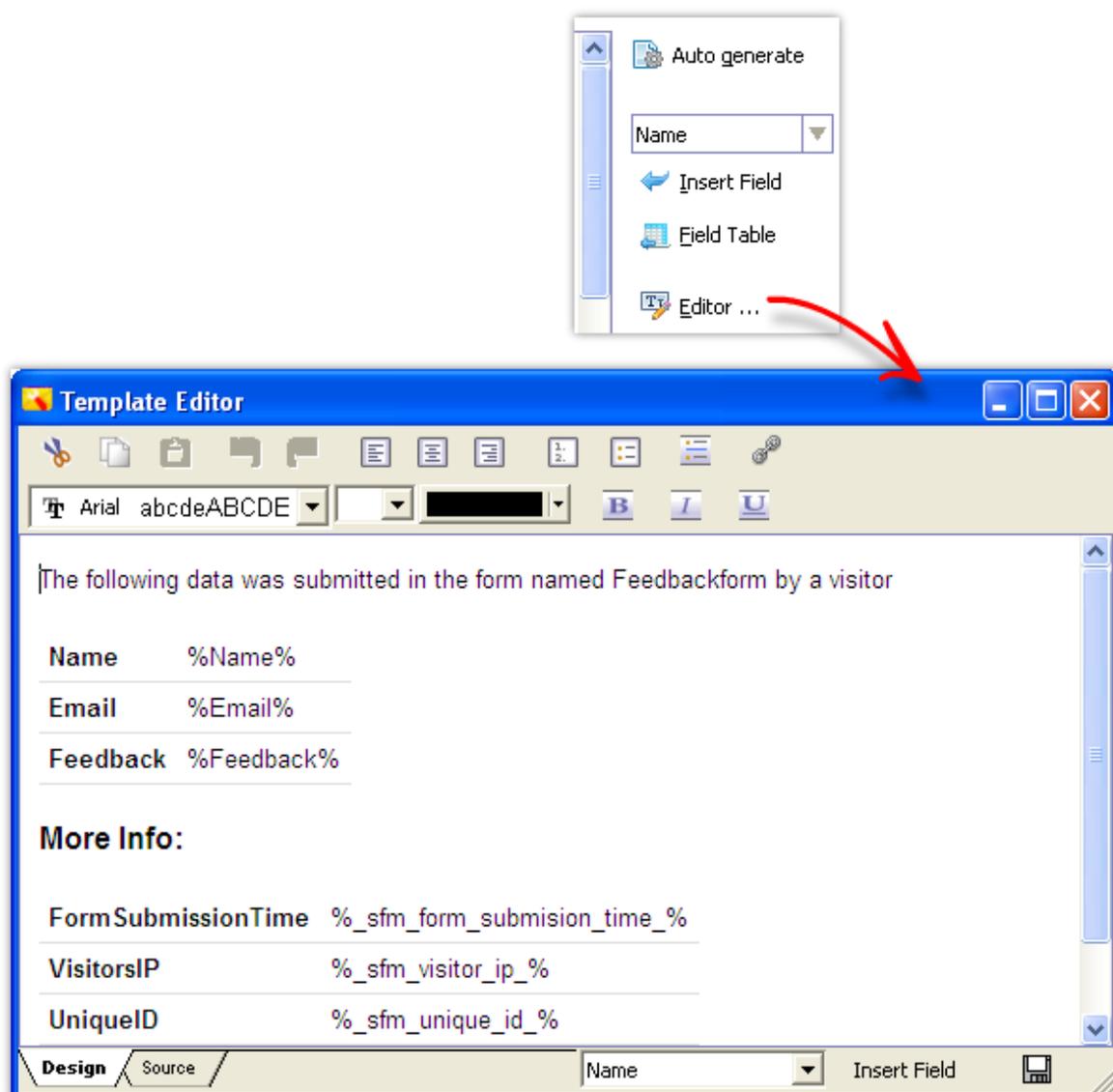
You can add more than one email addresses by following the same procedure.

Editing the Email template

Press the auto-generate button to create an initial template. The auto-generate button opens a 'Field table' box where you can change the order and placement of the fields.

Edit using the Template editor

The in-built HTML editor helps you edit and format the email content. Press the Editor button to open the Template editor.



Field place-holders

You will notice in the auto generated email the field names marked like %Name% %Email% etc. These are the place-holders for the values entered in the form. If somebody enters 'John Smith' as name, then while processing, %Name% will be replaced with 'John Smith' .

In the editor, you can select a field and press 'Insert Field' button to insert the placeholder for the selected field.

Once you have completed editing, press the 'save' icon and close the editor.

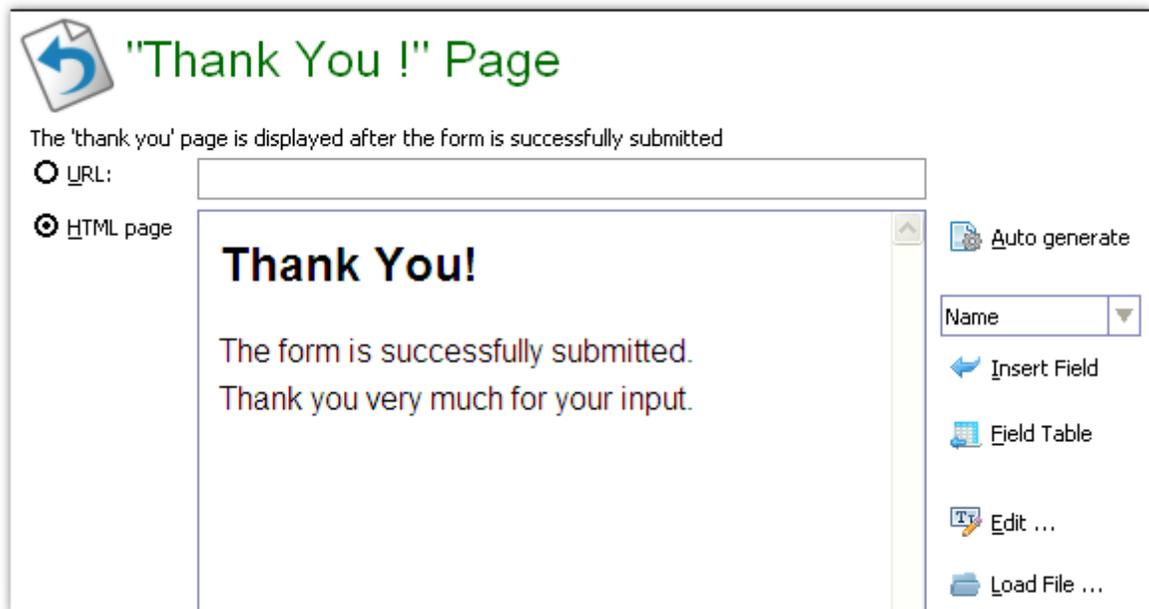
We have completed configuring the form-email. Press the 'Next' button

Prev: Select Form Processing Options | Next: Edit 'Thank you' Page

Step 6: Edit 'Thank you' Page

The "Thank you" page is displayed once the form is submitted successfully. You can configure the response page in this step. After the successful submission of the form, You can either

- Redirect to a URL or
- display a page.



Create the 'Thank You' page template

The quick way to create the 'Thank You' page is to use the 'Auto generate' button. Pressing the 'Auto generate' button generates a sample 'thank you' page.

The other alternatives are:

- Give a URL to redirect to. (Select the 'URL' radio button and enter the URL).
- Create and keep a 'Thank you' page. Load the file by clicking the 'Load file' button

You can press the 'Editor' button to open the template editor.

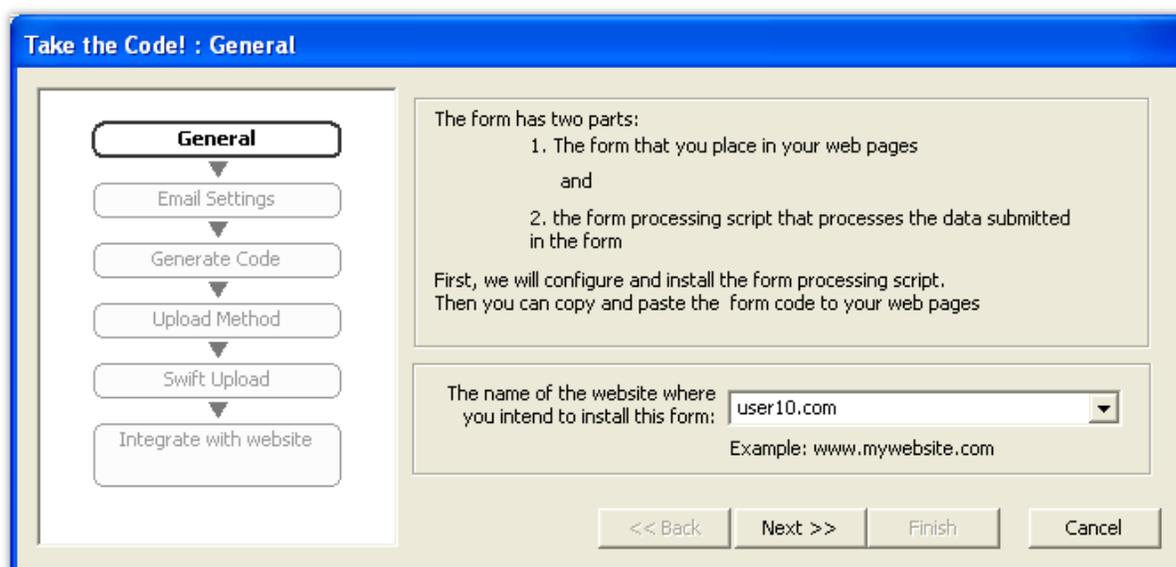
Press Next button to go to the next step

Prev: Edit Email Template | Next: Upload the Form

Step 7: Upload the Form

In this step we will install the feedback form on your web site. The 'Take the code' page helps you generate the code and then install the form on your web server.

Press the 'Take the code' button to open the wizard that takes you through the process



Page 1: General

Provide the name of the website where you would like to install the form and press 'Next'

Page 2: Email settings

You can modify the email settings if required. Press 'Next'

Page 3: Generate Code

Press the 'generate code' button to generate the code. After generating the code, Simfatic forms will display a message.

Press 'Next'.

Page 4: Upload Method

There are two ways to upload the form scripts to your website. Either use Simfatic Forms' Swift upload or use manual upload. The Swift upload is recommended because it is easy and safe.

[Note: the rest of the tutorial assumes that you have chosen 'swift upload' method.]

Page 5: Swift Upload

First you need to Add your web server to the list. For adding the server, press the 'Add Server' button. It will open a wizard that will collect the login information and the configuration of the web server.

After adding the server, choose the server from the list.

You can change the remote folder if required.

Press the upload button to upload the generated scripts to your web server.

Press the 'Finish' button to close the form installation wizard. Once you have installed the form successfully, you will see more options in the 'Take the code' page.

Press the 'Open form page' button to open the form in your web browser and try it out.

Integrate the form with your web site

You can embed the form in your web pages by copy-pasting the code. Press the

'Copy & Paste Form code' button. It will open a box giving the code for the form. Copy the code from the 'embed the form' tab and paste in your webpage/blog/CMS.

Done!

You have created and installed your first form.

Now you can keep watching the email inbox for new form submissions from your visitors.

Prev: Edit 'Thank you' Page | Next: adding more features to the form

2.2 Enhancements: adding more features to the form

Once you have successfully installed a simple form, it is easy to add more features to the form.

While making the form in Simfatic Forms, press the 'Save' toolbar button  (or alternatively, the menu item File->Save) to save your form design to a Simfatic Forms project file (The file will have extension .fwz). You can open the Simfatic Forms project file later, add more features and re-install the form again.

This section shows you how to add more features to the feedback form that we had created.

Next: Save the form submission data to a database table

Save the form submission data to a database table

It will be handy to have the form submissions saved to a database table. Let's see how we can have Simfatic Forms do that.

1. Open the Simfatic Forms project File

You should have saved the Simfatic Forms project file for the feedback form.

Open the file using Simfatic Forms.

2. Update the Form Processing Options

- Select the 'Form Processing Options' page.
- Enable the option 'Save the form submission data to a file in the server'

3. Select the fields to save

Press Next button to reach the 'Save to Database' page. You can select the fields to be saved this page. Leave all the fields selected and press the 'Next' button.

4. Install the form again

- Select the 'Take the code' page
- Ensure that the correct server is selected.
- Press the 'Re-Install form' button.
- follow the steps in the 'take the code' wizard

Viewing form submissions

Saving to a database table lets you view the form submissions online.

Switch to the form maintenance page



Press the Form Admin Page button.

You can view the form submissions in the Admin page (that opens in the browser)

There is an option to download the form submissions too.

Next: Send an auto-response to your visitor

Send an auto-response to your visitor

A visitor comes to your web site; takes time to submit your form. It is a great idea to send a response immediately. Using Simfatic Forms it is very easy to send a personalized auto response.

Update The Form Processing Options

- Open the Simfatic Forms project file that you had saved
- Select the 'Form Processing Options' page.
- Check the option 'Send an auto-response to the visitor who submitted the form'
- Press 'Next' to reach the 'Configure auto-response email' page

Create a response message

You will see 'Name field' and 'Email field' boxes that contains the list of elements in the form.

- For the 'Name field' select the element 'Name'
- For the 'Email field' select the element 'Email'
- Provide the Subject field as Thank you %Name% !

The %Name% will be replaced with the visitor's name when the email is sent.

- Enter the following as the text as the Email content.

```
Hello %Name%!
```

```
Thanks for visiting our website and submitting your  
feedback.
```

We will be in touch with you soon!

Regards

Webmaster

Re-install the form

- Select the 'Take the code' page
- Ensure that the correct server is selected.
- Press the 'Re-Install' button.

Test the Form

Submit the form and give your own email address.

Check your email inbox for the auto response!

Part



How To ...

This section contains 'How to' articles on using Simfatic Forms in certain situations. Look in this section when you want a specific task to be done using Simfatic Forms.

3.1 How to save form submissions to a file

You can let the Simfatic Form Processor save the form submissions to a file. The file is saved in csv (comma separated values) format. The fields in the form are saved in the file separated by comma(.). Individual submissions are separated by a new line.

Example

Suppose our form has three fields: Name, Email and Comments. The CSV file after a few submissions will look like this:

```
"John","john@hisdomain.com","no comments"  
"somebody else","they@their-domain.com","good web site"  
"Sara","sara@yahoo.com","your web site need optimization"
```

Many applications, like Microsoft Excel, can open csv files.

Select 'save to file' option

In the 'Form Processing Options' Page, select 'Save the data from the form submissions to a file in the server' check box

'Configure Form Submission Data File' page

The 'Form to File' page lets you select the fields to be saved to the form submission data file.

The right-side list shows the fields to be saved. By default, all the fields in the form are added to the 'to be saved' list.

Use the add/remove buttons to select the fields to be saved.

You can re-arrange the order of the fields in the file using the Move Up/Move Down buttons.

The field in the top of the list appears left most in the CSV file and other fields follow in the order.

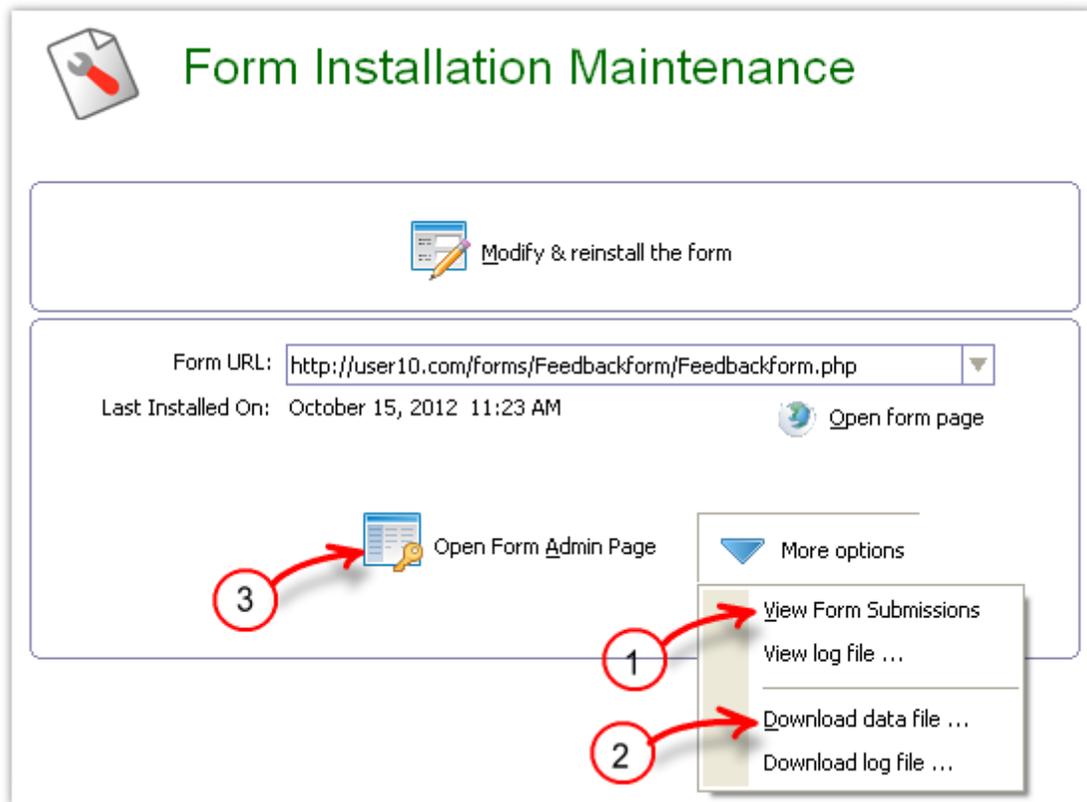
The data file location

While you install the form, Simfatic Forms configures a form data folder. The form data csv file is save in a form data folder.

Viewing the form submissions

After installing the form on the web server, make a few sample submissions in the form. Open the Simfatic Forms' project file (file extension .fwz) for the form.

By default, you will be taken to the Form maintenance page.



You can open the form maintenance page by pressing the toolbar button as well.

Select the 'View form submissions' option(1). This will display the form submissions in a data viewer.

Downloading the data file

Select the download datafile option (2) to download the CSV file.

 You can download the form submission data file from the Form Admin Page as well. Press the Form Admin Page button (3) to open the Form admin page in the browser. Login to the page and click on the link to download the CSV file.

Forms with file attachments

You can have file upload boxes in your form that lets your visitor upload files along with the form submission. If you have opted to have 'save to file' option for a form that has file upload box, the uploaded files are saved to a sub-folder inside the formdata folder.

The CSV file will contain the name of the file that was submitted and the actual file will be present in the upload folder.

 Note: Simfatic form processor prevents overwriting even if files with the same name are submitted.

You can choose to download the attachments while downloading the CSV file.

3.2 How to send an auto response

You can send a response email to the visitor who submitted the form.

Your form should have a name and email field to collect the name and the email address from the visitor.

Select auto response option

In the 'Form Processing Options' page, select 'Send an auto response to the visitor who submitted the form' check box.

This adds a step 'Response Email' in the wizard.

'Configure Auto response Email' page

The 'Auto-response' page lets you edit the auto response email template.

Select the Name field from the drop down where your visitor submits her name. Similarly, select the email field where the visitor's email address is submitted.

Using the field values from the form in the email

You can embed the values from the form submission in the email body.

For example, if there is a field 'Name', you can have the email body as

```
Hi %Name%,
```

```
Thanks for the form submission.
```

```
Webmaster
```

While sending the auto response to a visitor who submitted his name as John, the email body will be

```
Hi John,
```

```
Thanks for the form submission.
```

```
Webmaster
```

Insert Field

Select the field that you want to embed in the email body from the drop down and press 'Insert field' button.

This inserts a code of the form `%field_name%` in the email body. This code is replaced by the actual value for the field from the form submission.

Install and Test

Once the auto-response template is configured, generate code, install the form on your web server, and test by making a sample submission.

3.3 How to make changes to an already installed form

You may want to make changes to a form that is already installed. (Please note that this refers to forms that are created using Simfatic Forms only)

You can add more form elements, add new features and install the form.

Here are the steps:

- Open the The Simfatic Forms project file (the file with extension .fwz)



- Make the required changes
- Proceed to the 'Take the code' page
- Press the re-install the form button

What happens to the existing form submission data?

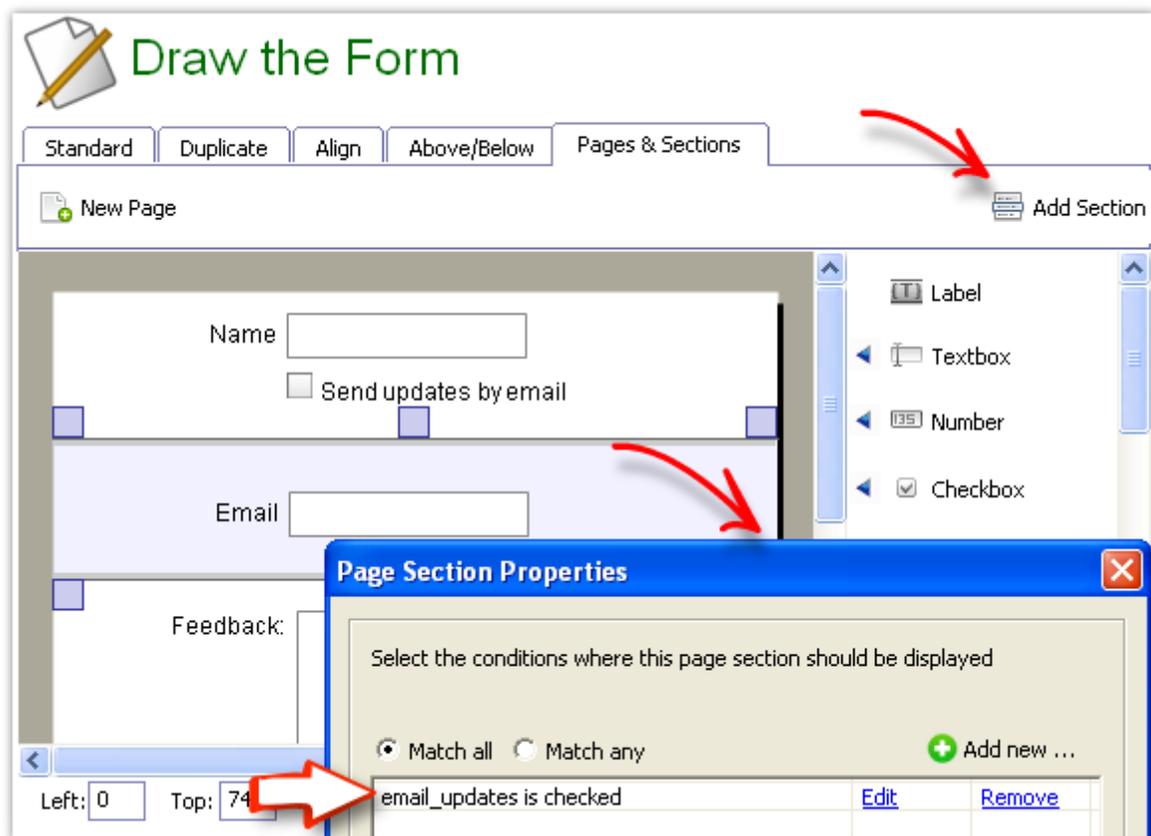
You already may have got several submissions to the form. Re-installing the form does not affect the already existing data. If you had chosen to save the form submissions to a file or if you had selected saving form submissions to database, the previous form submissions will remain intact even after re-installing the form.

3.4 How to display elements conditionally

You can simplify long forms by displaying certain elements only on certain conditions.

Simfatic Forms supports conditionally displaying a section.

Here are the steps:



- First, place a form section on the form by pressing the 'Add Section' button. (Pages & Sections -> Add Section)
- Place your form elements on the section
- Double-click on the form section. The section property dialog box opens.
- In the form section property dialog box, add the conditions.

- The form section will be displayed only when the conditions are true.



You can use calculation fields in conditions. That means, for complex conditions, convert the condition to a formula and use that calculation field in the condition.

Hidden fields can be used in the condition as well. Combine that with values passed from URL, you have forms with conditional sections depending on the URL.

See Also:

'Draw the form' Page

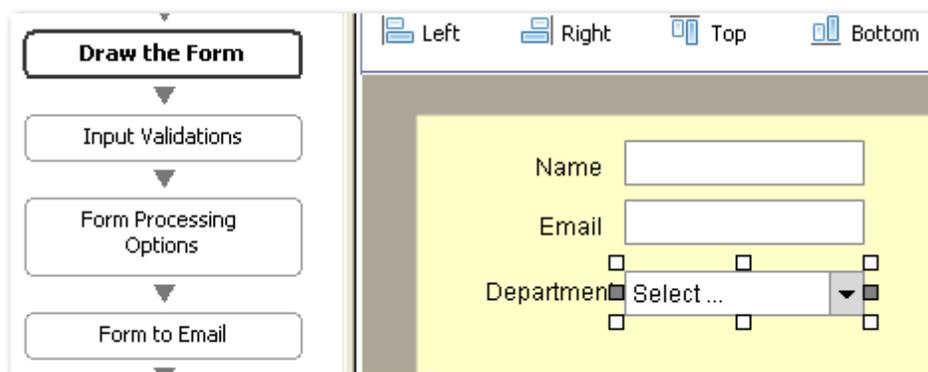
3.5 How to email based on a selection

It is often required to send the form submission email based on a selection in the form.

For example, the user selects from a "department" dropdown list and the email is sent only to that department.

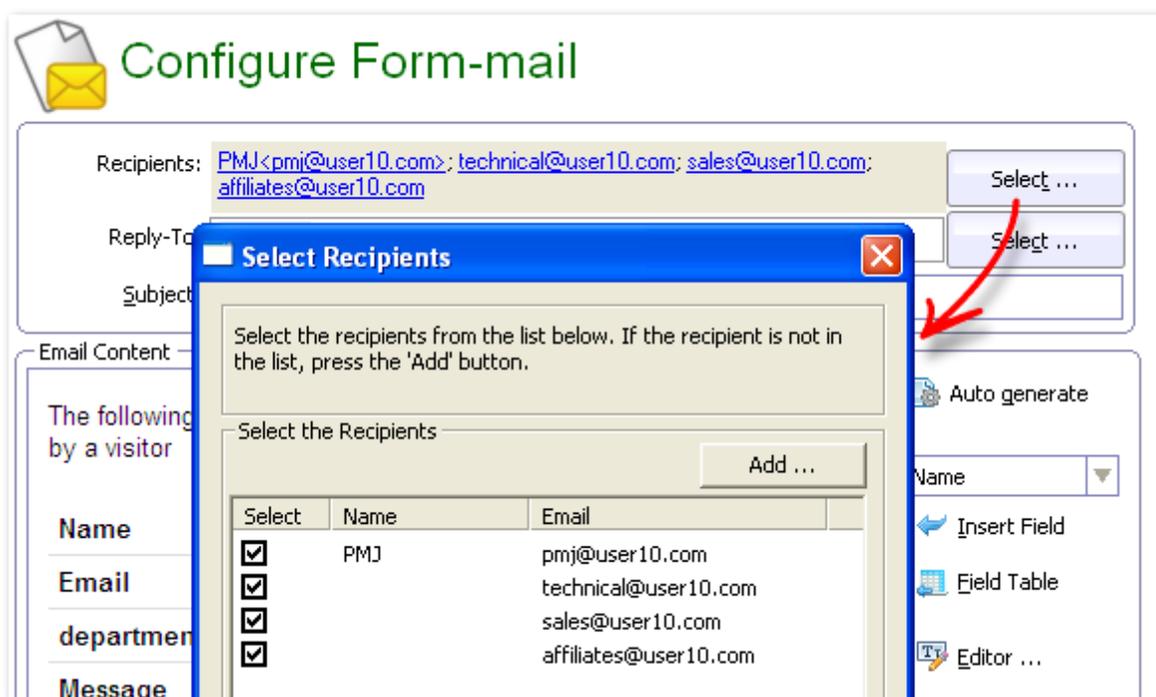
Here is how to do that in Simfatic Forms.

First, build the form and add the required elements in the "Draw the form" page.



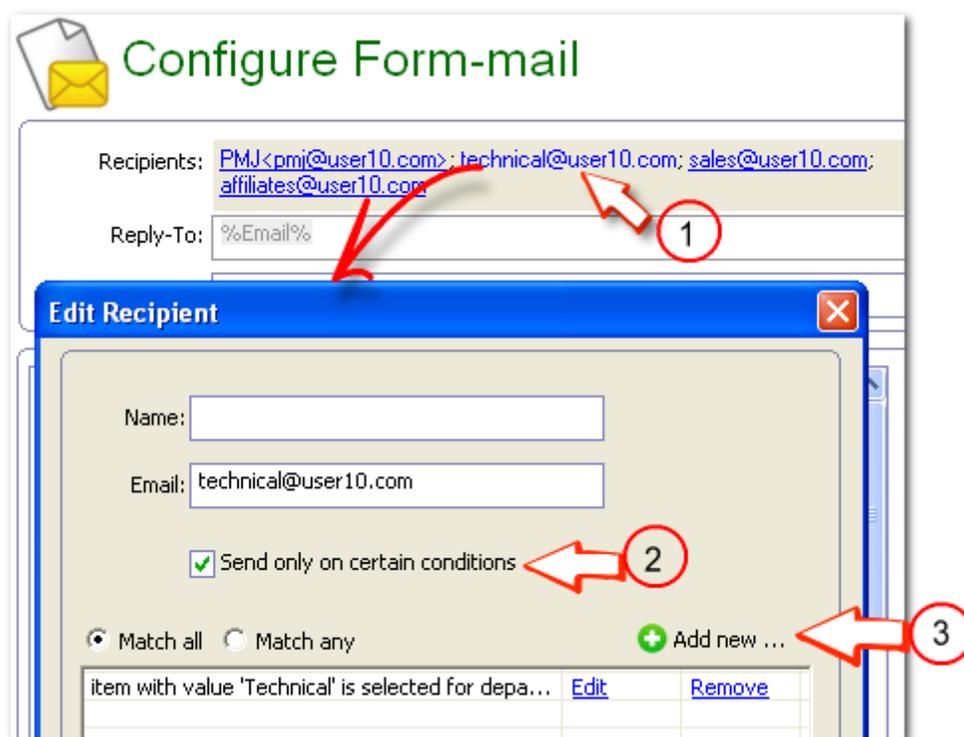
build the form

In the "Form to email" page, add all email addresses.



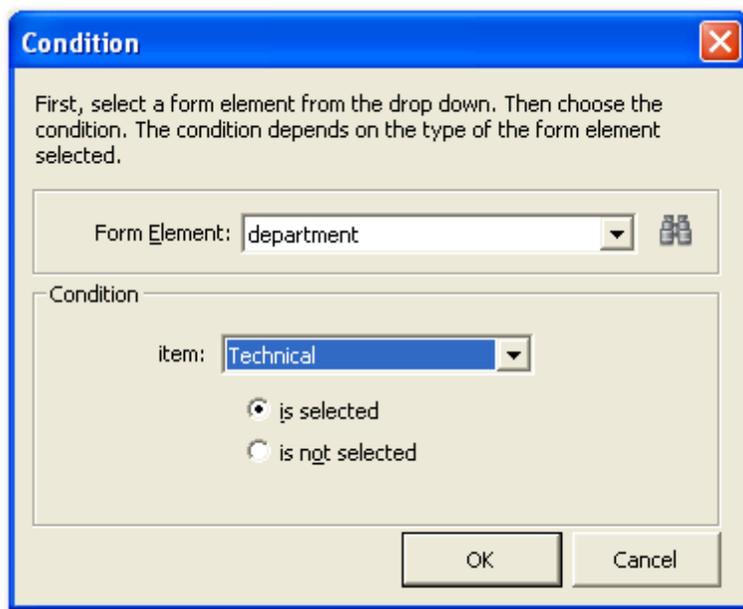
add all email addresses in the recipients list

Now click on each email address and select the condition to send the email



Click on the email and select the condition

For example, the department should be 'Technical' if the form submission should be sent to technical@user10.com



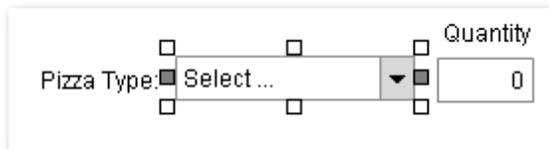
select condition

After adding the conditions, press next and proceed to uploading the form.

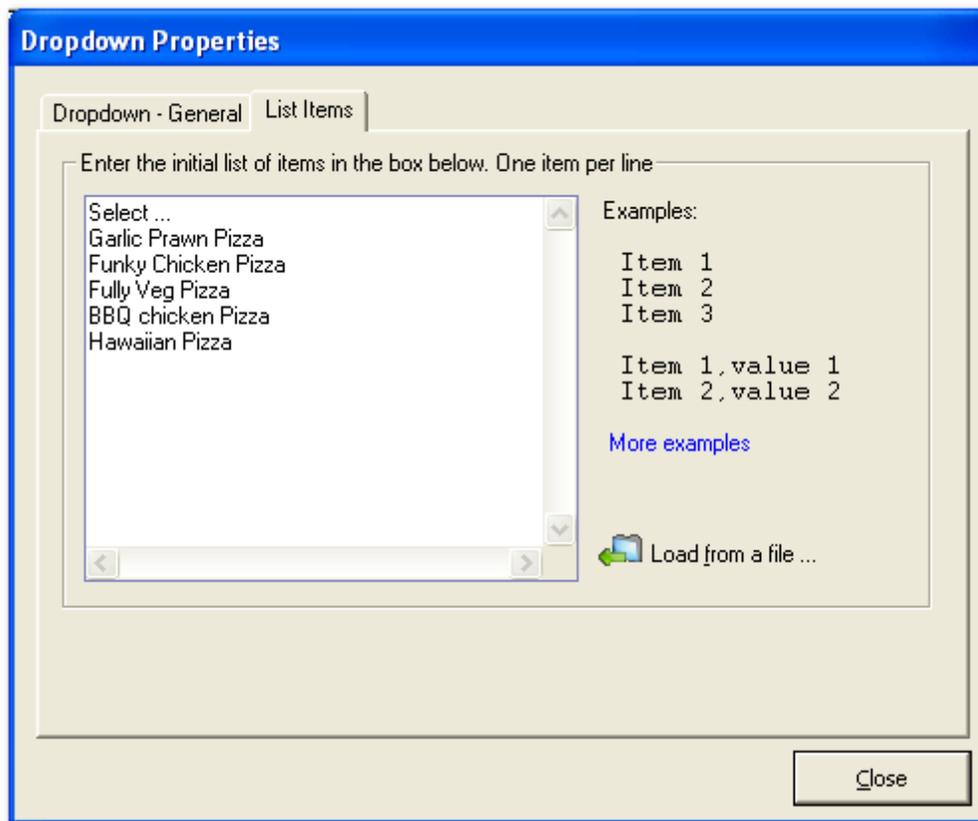
3.6 How to create a simple calculation field

Imagine you want to create a simple order form with item quantity and price. Here is how to make one quickly.

We have a pizza order form. A dropdown with pizza type and a "quantity" box to enter the quantity

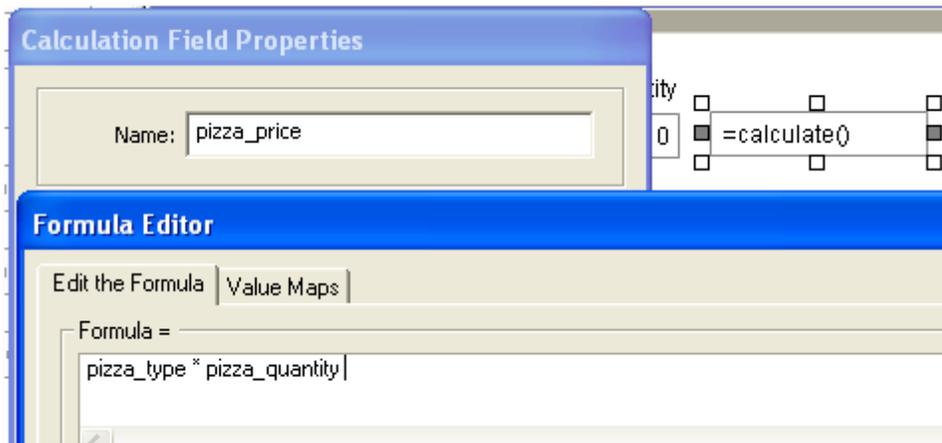


Here are the items in the 'pizza_type' drop down list:



The pizza price varies with the pizza type . Garlic Prawn Pizza is \$10 whereas Funky Chicken Pizza is \$12

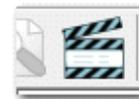
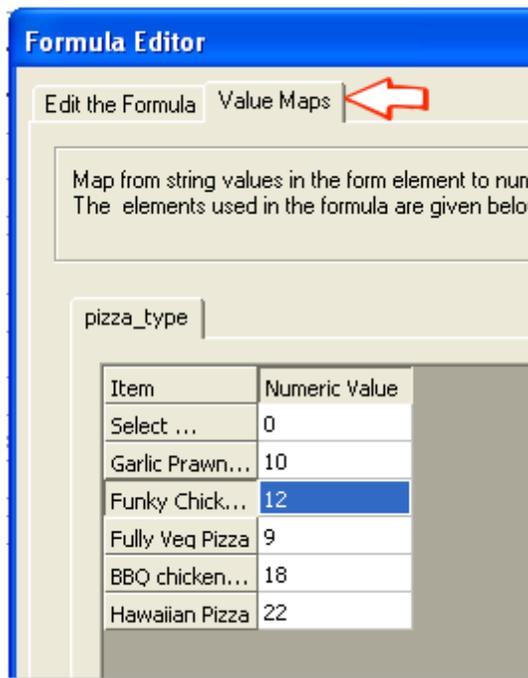
First, we add a calculation field



Notice the formula: $\text{pizza_type} * \text{pizza_quantity}$

Now we need to map from the 'pizza_type' dropdown list items to its price. We can do the mapping in the

'Value maps' tab:

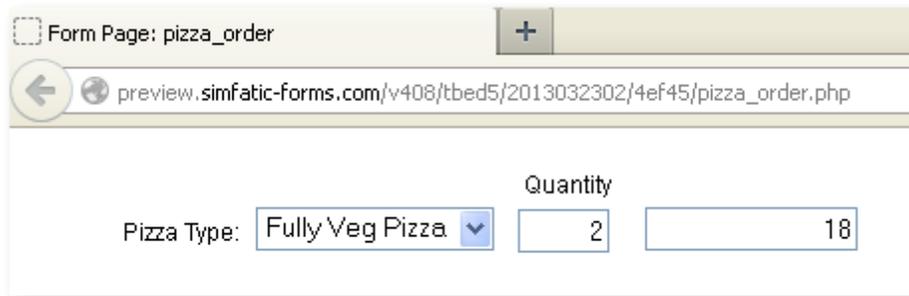


Now we have the calculation ready. Press the Preview button to see it working

The formula: `pizza_type * pizza_quantity` works this way:

Suppose the user selected "Fully Veg Pizza" the corresponding numeric value is taken from the value map (it is 9)

and then multiplied with the value in the `pizza_quantity` :



Value Maps

When you are using a selection (radio group, drop down list) in a formula, you have to map the selected value to a numeric value. Simfatic Forms' value map makes this easy. First enter the formula . If your formula contains a selection, the "value maps" tab will have mapping for such selection fields. Just provide the numeric value to be used in the calculation.

Now that the basic step is done, we can add a submit button and proceed to build a full fledged order form

Download the form template here:

[order form template](#)

More info:

[Calculation Field Reference](#)

Calculation field : function reference

Formula editor

How to create an online calculator

3.7 How to create an online calculator

The calculation field feature in Simfatic Forms can be used to do complex calculations live.

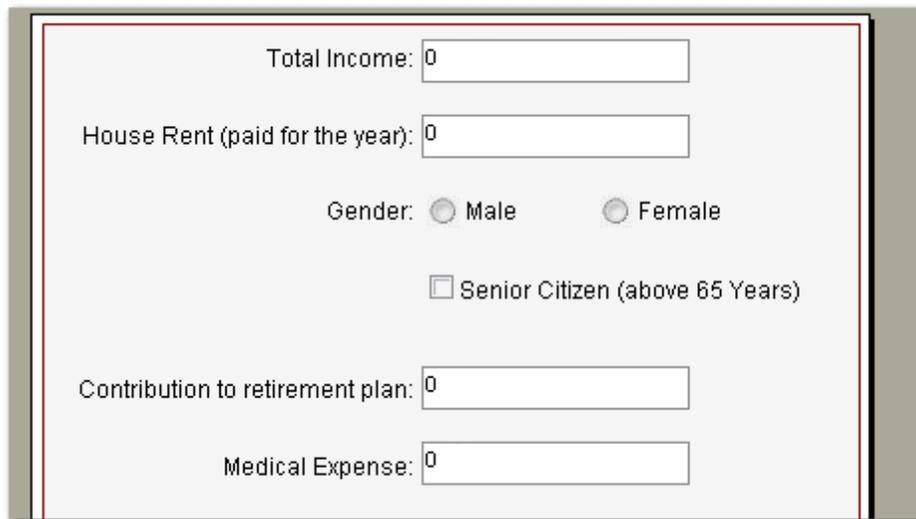
This article shows you an example that breaks a complex formula and logic for a tax calculation.

Note that the rules used are fictitious. You can assume any currency or any country.

Having said that, here are the rules of the land:

- You have got the total yearly income
- Contribution to retirement plan can be deducted (upper limit 10,000)
- The first 100,000 of the total income is not taxable; 30% tax for the rest
 - Example: if the income is 100,000 then there is no Tax.
 - If the income is 110,000 then tax should be paid for 10,000. That is, tax = 30% of 10,000)
- For senior citizens and women, there is no tax for the first 150,000
- There is a 10% tax rebate for House Rent (upper limit 100,000)
- There is a 20% tax rebate for Medical Expenses (upper limit 15,000)

Here, we have the form:



The screenshot shows a form with the following fields and options:

- Total Income:
- House Rent (paid for the year):
- Gender: Male Female
- Senior Citizen (above 65 Years)
- Contribution to retirement plan:
- Medical Expense:

It is always better to break a big formula into smaller ones.

Part I : Taxable Income

To calculate the taxable income, we have to deduct the 'contribution to retirement plan' from the total income.

```
taxable_income = total_income -  
retirement_plan_contribution
```

But, there is an upper limit to the `retirement_plan_contribution`. Simfatic Forms provides several functions. The `min()` function returns the minimum of the inputs. So, can make it this way:

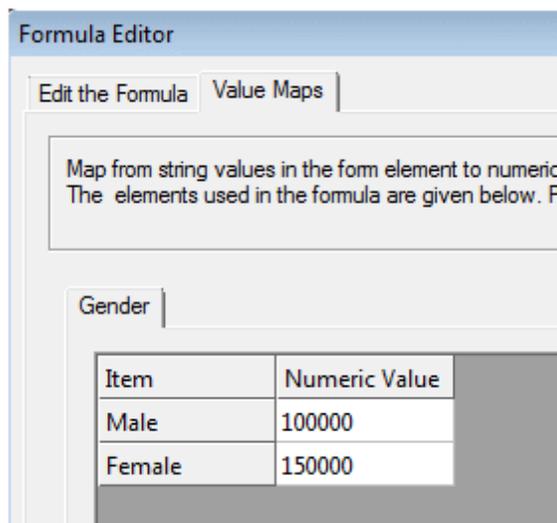
```
taxable_income = total_income - min  
(retirement_plan_contribution, 10000)
```

Now, there is a discriminating minimum income limit. We take the input from the 'Gender' radio button. Here is the updated formula:

```
taxable_income = total_income - min  
(retirement_plan_contribution,10000) - Gender
```

When we add a radio button or dropdown list to the formula, we get a new tab in the 'Value maps' tab of the formula editor. This is to map the selection to its numeric value.

For example, if the 'Gender' is Male, we need to use 100000 and for Female, 150000.



Value Maps

When you are using a selection (radio group, dropdownlist) in a formula, you

may have to map the selected value to a corresponding numeric value. Simfatic Forms' value map makes this easy. If your formula contains a selection, the value maps tab will have mapping for each of such inputs. Just provide the numeric value to be used in the calculation.

There is, an additional rule that Senior citizens also get the 150000 limit. So, we update the formula such that if not SeniorCitizen, take the Gender based limit. If SeniorCitizen, then always deduct 150000. Here is the updated formula:

```
taxable_income = total_income - min
(retirement_plan_contribution,10000 )
- !SeniorCitizen * Gender - SeniorCitizen *
150000
```

Notice that the check box 'SeniorCitizen' returns 1 if checked and 0 if not checked.

Part II : Calculate Tax

We now has the final taxable income. Tax is 30% of the taxable income. Simfatic Forms has a function perc_of() to calculate percentage.

```
tax_before_deductions = perc_of(taxable_income,30)
```

To reach the tax payable, we need to deduct the tax rebates.

```
tax_payable = tax_before_deductions
- perc_of( min( HouseRent ,100000) ,10)
```

```
- perc_of( min( medical ,15000),20)
```

We have got the final tax amount. But there is a problem; the tax can become negative if the income is low !

I know no government that pays tax to the citizen 😊

We can fix this by having the formula below

```
tax_displayed = min(tax_payable,0)
```

You can use hidden calculation fields to keep the intermediate results. In this calculation, taxable_income and tax_payable can be hidden; displaying only tax_displayed to the user.



During the design phase, keep the intermediate results visible so that you can see how the calculation goes. Once the calculations are working, you can make them hidden.

See Also:

Calculation Field

Calculation field : function reference

Formula editor

3.8 How to load list box items from a file

You can load the list box items from a simple text file. This will be handy if you want to load same list of items in many different forms. (For example, the list of countries.)

File Format

The format of the file is simple. One item pre line in a plain text file. Example:

```
Item 1
Item 2
Item 3
Item 4
```

If you want to have different name and value for the items, provide the value separated with a comma.

Example:

```
Item 1, Item1 Value
Item 2, Item2 Value
Item 3, Item3 Value
Item 4, Item4 Value
```

Cascaded Lists

For Cascaded lists, the format is like this

```
ParentItem 1 >
{
```

```
Item 1.1
Item 1.2
Item 1.3
Item 1.4
}
ParentItem 2 >
{
Item 2.1
Item 2.2
Item 2.3
Item 2.4
}
Common Item 1
Common Item 2
Common Item 3
```

'ParentItem 1' is the item value of the parent list for which the Items inside the braces '{' and '}' are to be shown. For example, When 'ParentItem 1' is selected, the cascaded list will show Item 1.1, Item 1.2 etc Whereas when ParentItem 2 is selected, Item 2.1, Item 2.2 etc are shown.

Loading from a file

In the Dropdown list properties box, List Items tab, press the 'Load the items from a file' button.

A file selection dialog box opens. Browse and select the file containing the list box items. This will fill the list of items with the items from the file.

See Also:

Filling the List items

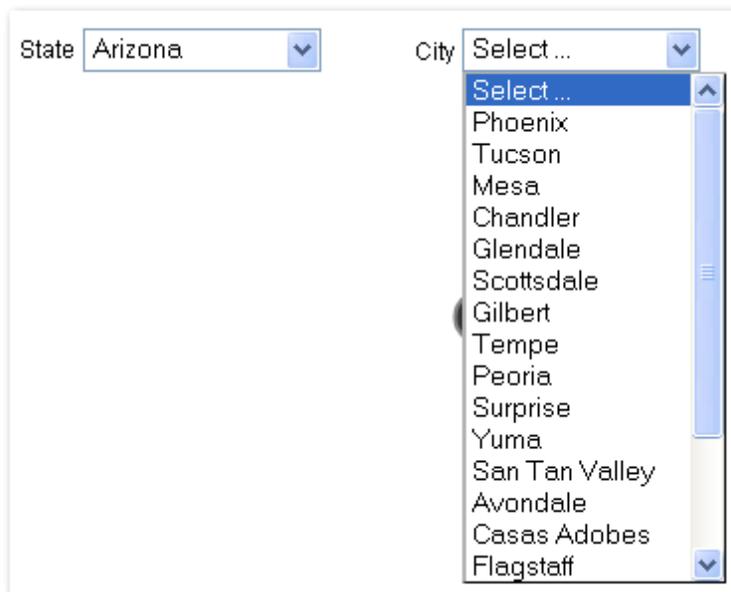
Dropdownlist properties

Multi-select box properties

3.9 How to create a cascaded list

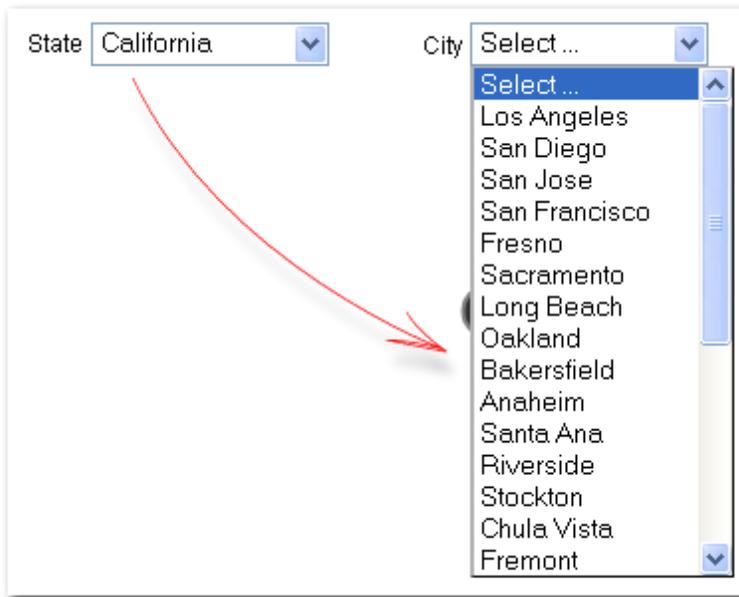
Suppose you want to load the items in a list based on another list. Cascaded list is the solution. Cascaded list is very useful if your form involves complex selections.

For example, a 'City' list that loads based on a 'State' list



The image shows a web form with two dropdown menus. The first menu, labeled 'State', has 'Arizona' selected. The second menu, labeled 'City', has a dropdown menu open showing a list of cities in Arizona: Phoenix, Tucson, Mesa, Chandler, Glendale, Scottsdale, Gilbert, Tempe, Peoria, Surprise, Yuma, San Tan Valley, Avondale, Casas Adobes, and Flagstaff. The 'Select...' option is highlighted at the top of the dropdown menu.

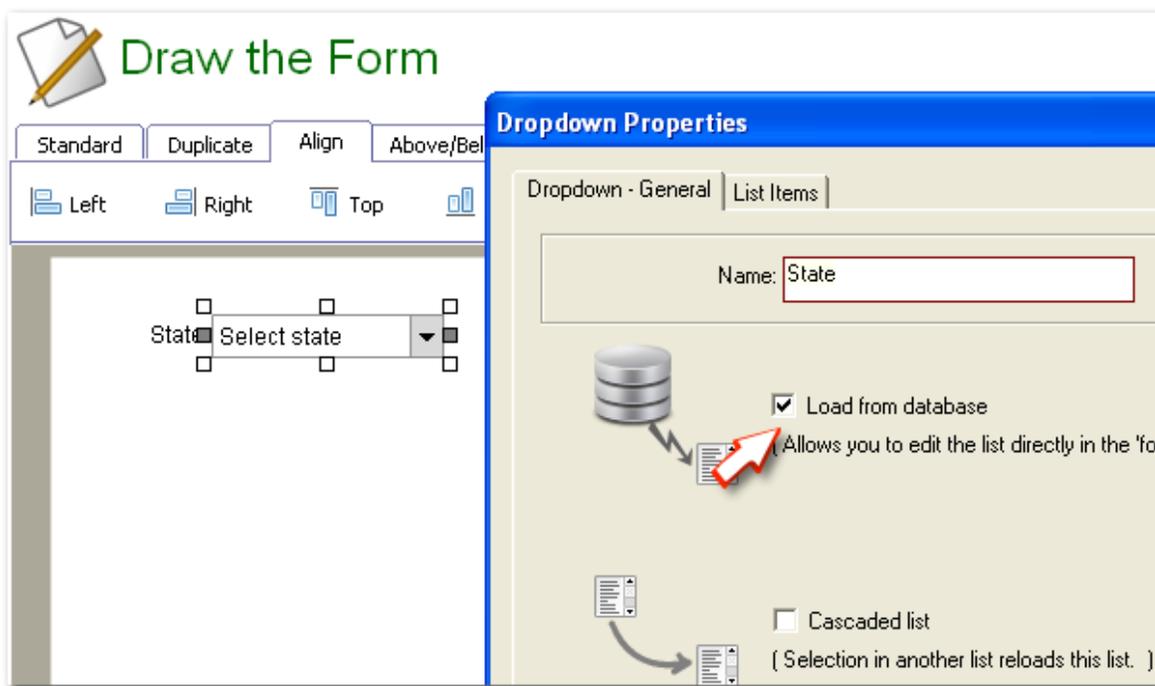
Cities in arizona



The cities reloaded

Creating a cascaded list.

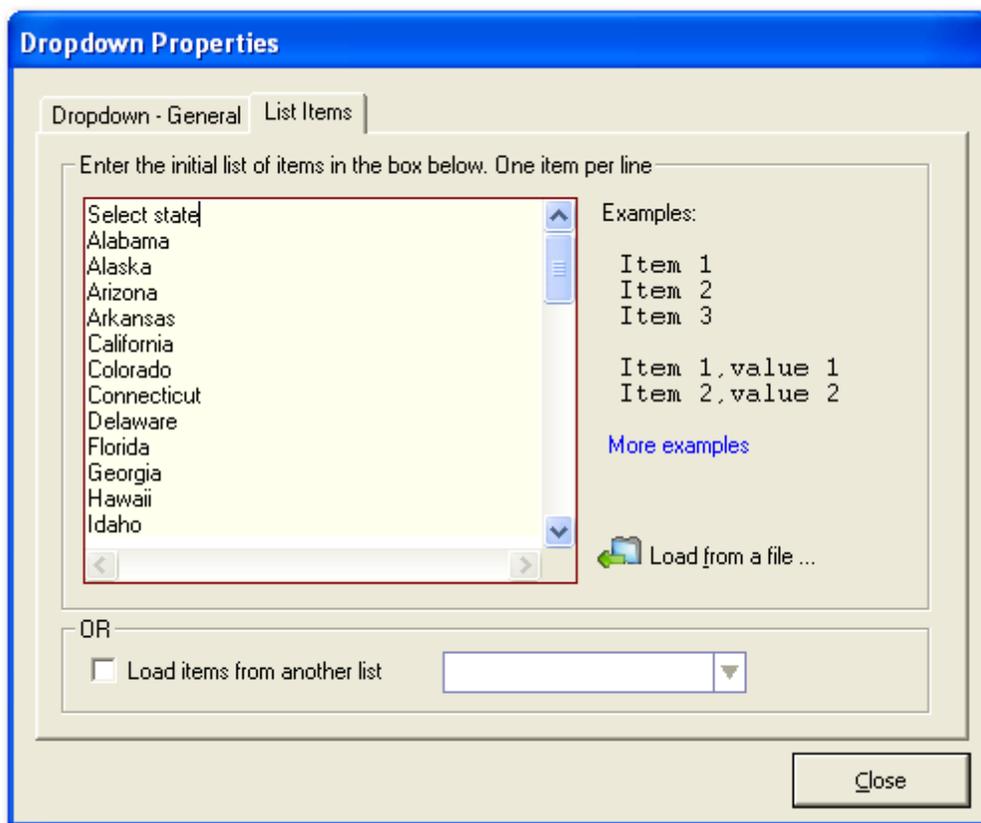
First create the parent list (The 'State' list in this example)



Parent list

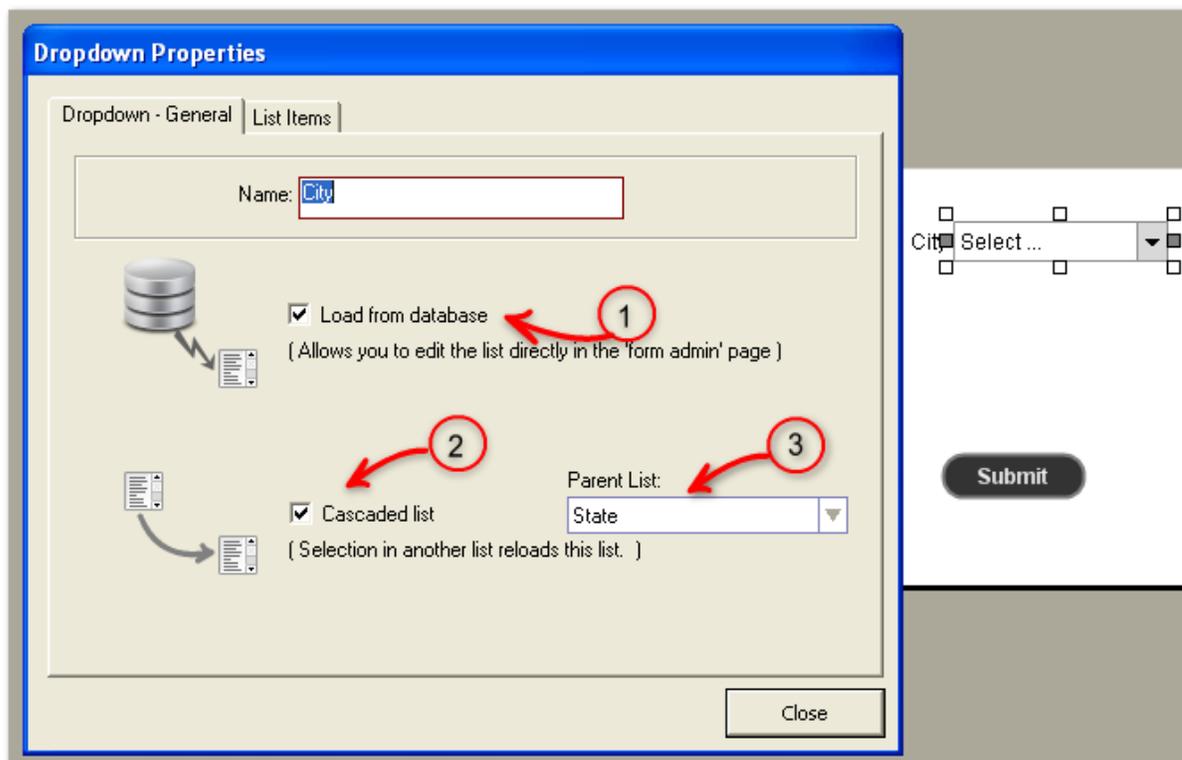
Enable "Load from database" option.

Add the items. List of U.S. states in this case :



Add items in the list

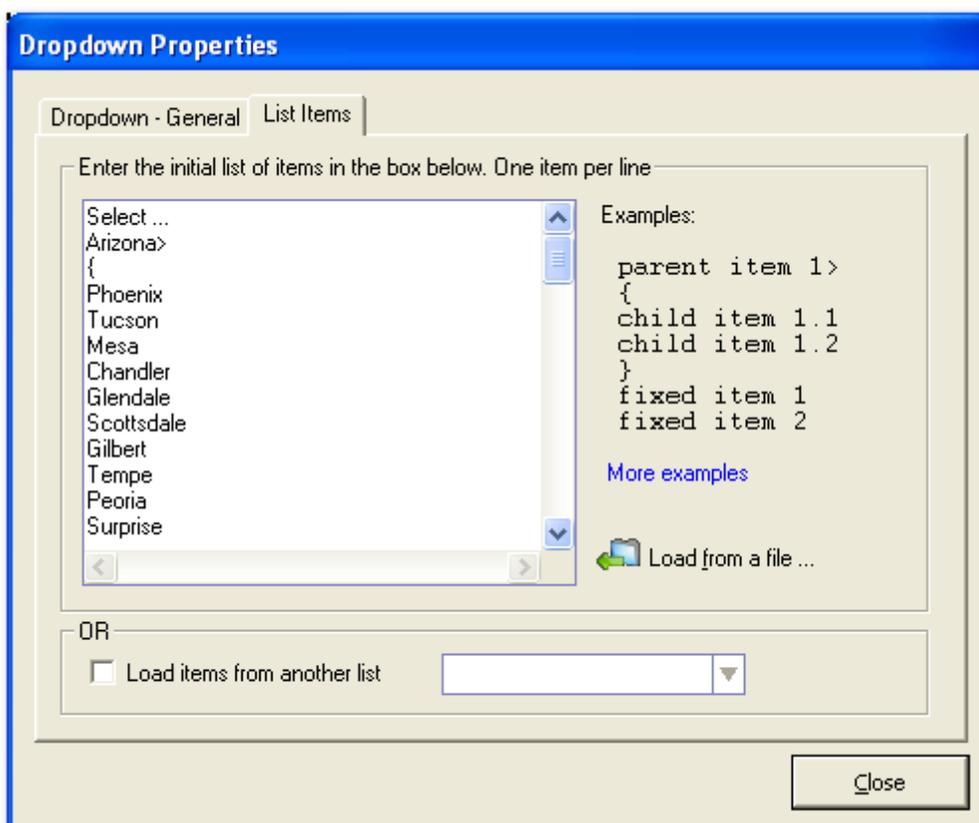
Create the 'child' list - a dropdown list of the cities in our example.



Create the child list

Choose the 'Load from database' option and the 'Cascaded list' option. Then choose the parent list.

Now we will add the list items.



List items follow a format

Enter the child list items in this format

```
Parent_item1>
{
child item 1
child item 2
child item 3
}
Parent_item2>
{
child item 1
```

```
child item 2  
child item 3  
child item 4  
}
```

and so on

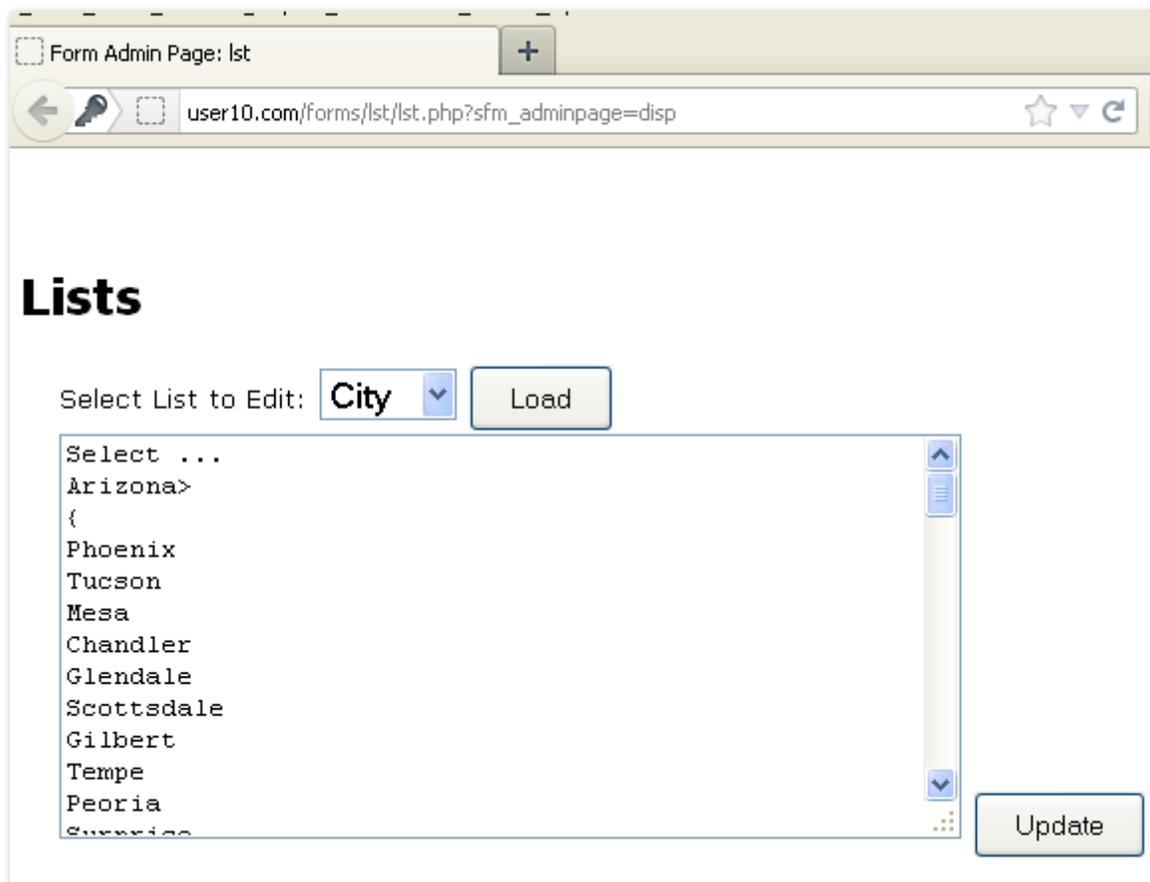
For the State->City example:

```
Arizona>  
{  
Phoenix  
Tucson  
Mesa  
Chandler  
Glendale  
}  
California>  
{  
Los Angeles  
San Diego  
San Jose  
}
```

Now proceed with uploading your form.

 Cascaded list requires database support. The lists are saved in to the database and are loaded dynamically. So, in the "Take the code" step, it will ask for a MySQL database login even if you have not selected any other database option.

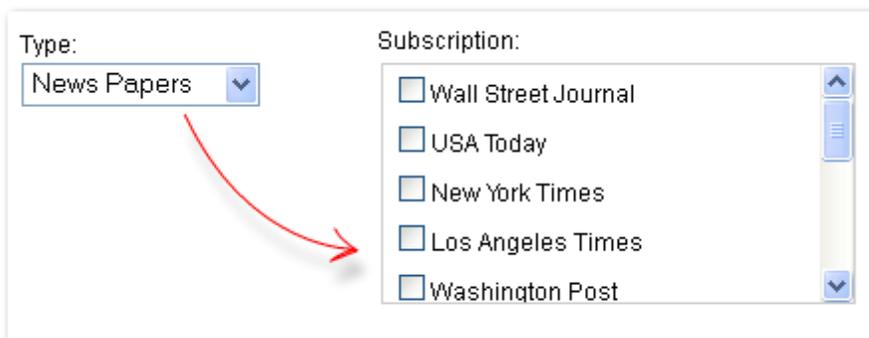
 You can edit the cascaded lists online in the Form Admin Page



Edit cascaded list online



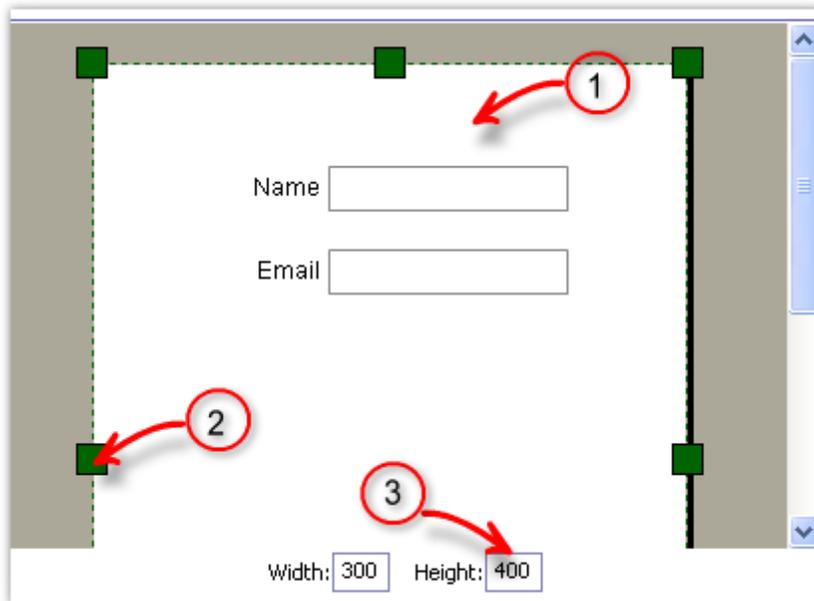
You can even build a cascaded multi-select list!



The screenshot shows a form with two sections. The first section, labeled 'Type:', contains a dropdown menu with 'News Papers' selected. A red arrow points from this dropdown to the second section, labeled 'Subscription:'. The 'Subscription:' section is a multi-select list with five items, each with an unchecked checkbox: 'Wall Street Journal', 'USA Today', 'New York Times', 'Los Angeles Times', and 'Washington Post'. The list has a scrollbar on the right side.

cascaded multi-select

3.10 How to change the form page size



1. Click in the blank area of the form page

2. Drag the handles to resize.

OR

3. Edit the exact size

3.11 How to add a conditional validation

If you want to do a validation only on certain conditions, you can do that easily in Simfatic Forms. There is a built-in conditional validation feature.

The screenshot shows the 'Add Input Validations' dialog. On the left, a tree view shows the form structure: 'feedbackform' contains 'contact' and 'Other'. Under 'Other', there is a validation node 'required field [when: item wi]' with a green checkmark. A red circle '1' and an arrow point to this node. In the center, the 'Conditional Validation' section has a checked checkbox 'Validate only on the following condition:'. A red circle '2' and an arrow point to this checkbox. Below it, the condition text reads 'item with value 'Other' is selected for contact'. To the right of this text is a button labeled 'Condition ...'. A red circle '3' and an arrow point to this button. Above the conditional validation is an 'Error String' field containing the text 'Please fill in Other'.

- First, add the validation.
- Select the validation node (with the green tick) .
- On the left side, enable the validation condition.
- Press the condition button and select the condition



If you have a complex condition, use a hidden calculation field. You can

provide the complex condition as a formula. Then check that calculation field in the validation condition.

See Also:

'Add Input Validations' page

Validation condition dialogbox

3.12 How to add your own validations

If the validations provided by Simfatic Forms does not meet your need, you can add your own custom validation function.

The custom validation function need to be in JavaScript and runs only on the client side.

 For more information on how to add custom server side validation, Please see DoValidate() in Extension Modules

Steps for adding a custom validation function

You can add the custom validation function in the 'Input Validations' page.

1. Click the 'Validation Options' Link

Clicking the validation options link displays the validation options page on the right side.

2. Check 'Custom validation function'

There is a check box 'Custom validation function'. Enable it.

3. Press the button 'Edit Custom Validation function'

Pressing the 'Edit Custom Validation function' button opens an editor where you can provide the JavaScript code for the validation.

4. Provide the custom validation code

Enter your validation code. If the validation fails, it should return an object containing the error message and the element object to which the error belongs to.

Example:

```
if(formobj.elements['Name'].value == 'Bob')
{
    return
    ({
        elementobj:formobj.elements['Name'],
        message:'Bob, you cannot submit this form!'
    });
}
return true;
```

 Make sure you return true if there is no error.

See Also:

'Add Input Validations' page

3.13 How to upload the form manually

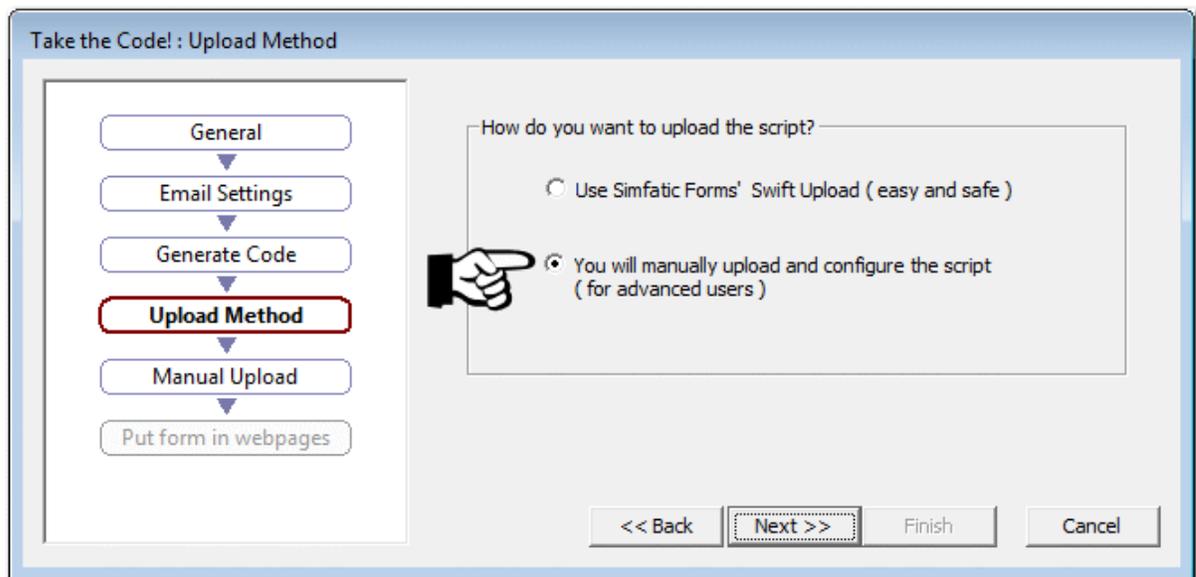
Some times, it may be required that you want to upload the form processor script manually to your web server. Simfatic Forms provides the option to install the form on the server manually.

Here are the steps:

1. Select 'manual installation' in the 'Take the Code' wizard

In the 'Take the code' page, when you press the 'Take the code' button, it opens the wizard.

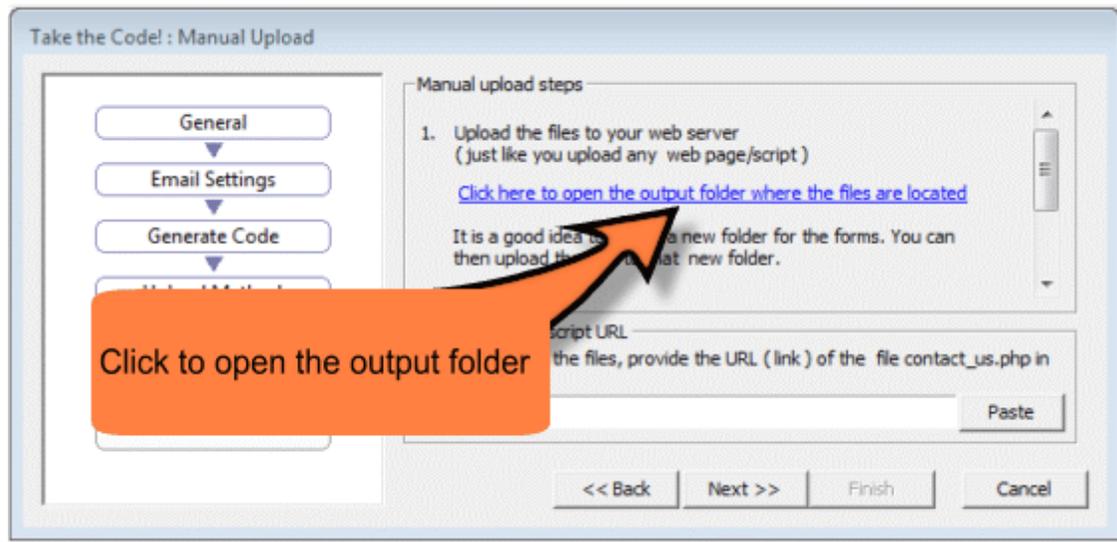
In the step 'Upload method', choose 'Manually upload and configure script' option.



Press 'Next'.

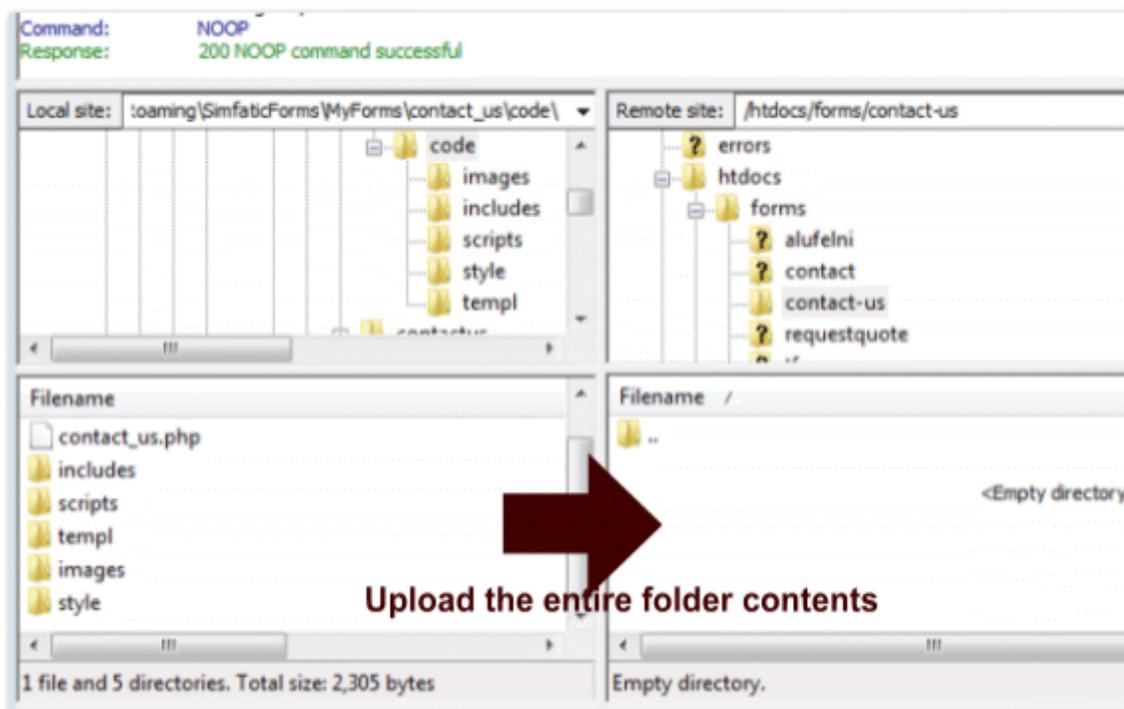
2. Upload the files manually

In the 'Manual upload' page, the instructions for uploading the files are shown.



Upload All the generated code to an appropriate folder on your web server.

Open your FTP client(if you are uploading through FTP). It is recommended that you create a folder for the new form. This makes it easy to manage the files and folders. Upload the entire output folder contents to the remote location.



3. Update formdata folder permissions

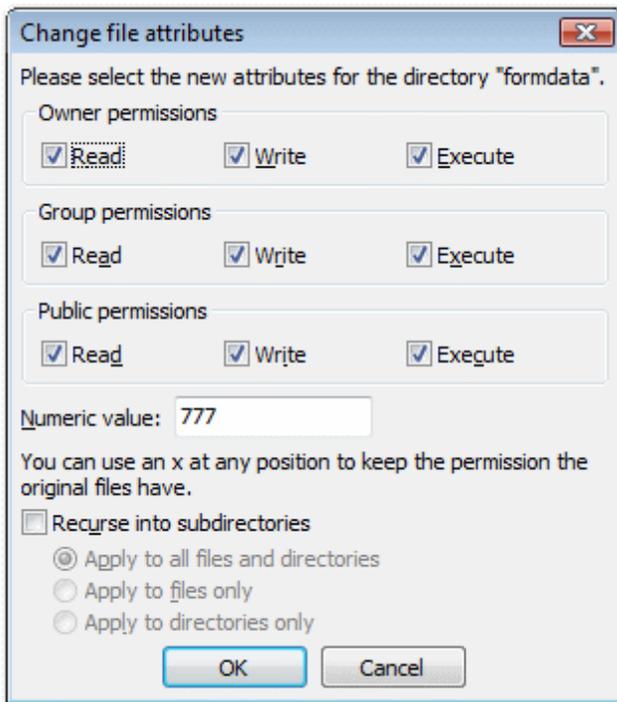
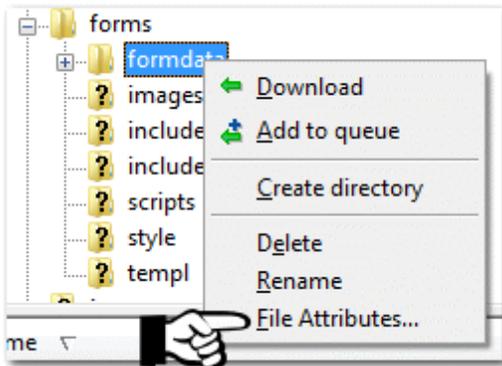
Follow the next step in the "manual upload steps" box. In certain cases, it will be to update the formdata folder permissions.

 This step is required only if the instructions in your manual upload page has a step to update the formdata folder permissions.

"formdata" folder is required only if you had selected features that requires saving to file (Logging, file uploads, saving to CSV file etc)

Form data folder is where Simfatic Forms keep files like the log file, the form data csv file etc. The script should have read and write access in the form data folder.

Most of the FTP clients lets you update the folder permissions. (right-click on the folder to open the context menu, that usually contains an option. Please refer the manual of your File uploader for details). The images below are that of the free FTP client FileZilla.



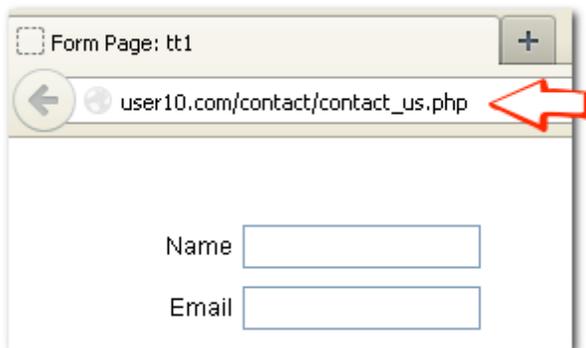
4. Update the form URL

Now you can check whether the form is online.

Enter the URL of the form in your browser. For example, if your form's name is contact_us and you uploaded the generated code to /htdocs/contact folder (htdocs being the folder where you publish your web pages)

the form URL will be:

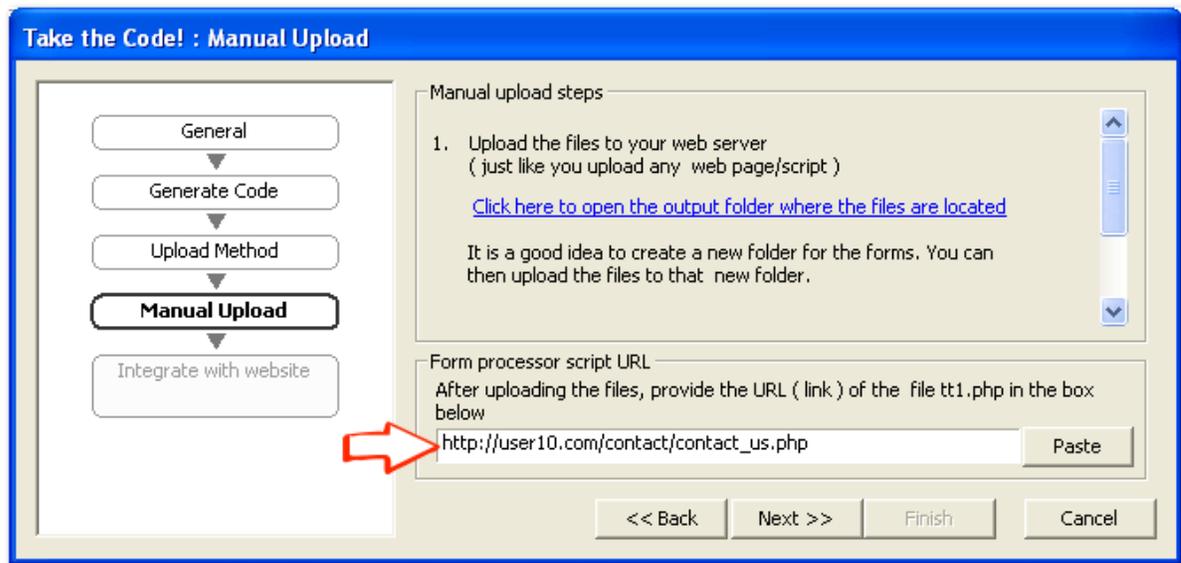
http://www.mywebsite.com/contact/contact_us.php



test the form url

If you are getting 404 not found error, then the URL you entered is wrong.

Copy the form URL to the Take the code->manual upload box:



update the URL

Press Next.

Simfatic Forms will test the URL. This will also trigger a set of validations in the script.

The script will validate to ensure that the form can function correct (like the SMTP login, database login etc)

The URL is required for generating the code to embed the form in your website as well.

Once the validations are complete, Simfatic Forms will show a 'success' message and the manual upload process is complete.

See Also:

Take the code Wizard

3.14 How to troubleshoot form errors

The code generated by Simfatic Forms runs on your webserver. It is possible that a configuration setting or other parameter can affect the functioning of the form. In such cases you have to troubleshoot the problem. This page provides some general tips on troubleshooting form errors.

Enable the troubleshooting options

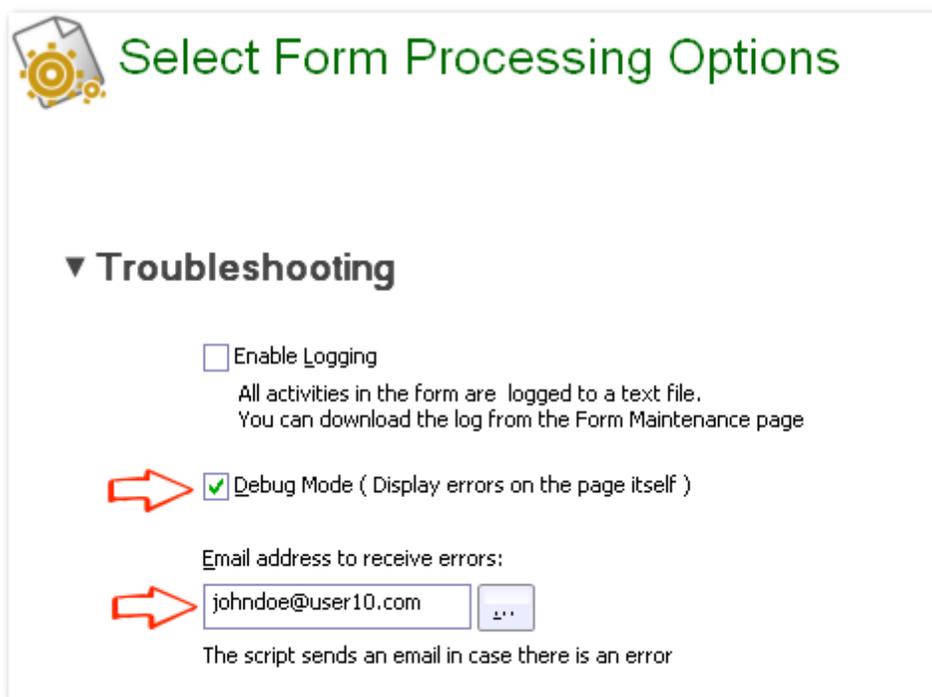
Debug mode

In the 'Form Processing Options' page, check the 'debug mode' option. This will enable displaying the errors on the page itself.

Once the form is working fine, you can uncheck the 'debug mode' option and re-install the form.

Email address to receive errors

In the 'Form Processing Options' page, provide your email address in the box labeled 'email address to receive errors'. If this option is enabled, Simfatic Forms form processor will send you an email that contains the details of the error.



It is recommended to always have your email address in the 'email address to receive errors'. This ensures that you are informed if there is any unexpected error.

 Once you have completed troubleshooting, you can uncheck the 'debug mode' and 'logging'. This helps in improving the performance.

Isolating the problem

If you have selected many features (form to email, save to database, save to file etc), it would be difficult to isolate what is causing the problem. Try unchecking the options in the form processing options page. Re-upload the form and test again.

3.15 How to install the form on a subdomain

Imagine your web server address is www.mywebsite.com and you have a subdomain subdomain.mywebsite.com. You want to install the form on the subdomain and access using URL <http://subdomain.mywebsite.com/myform.html>

In order to access files using a sub-domain, you upload files to a folder say, /subdomain/

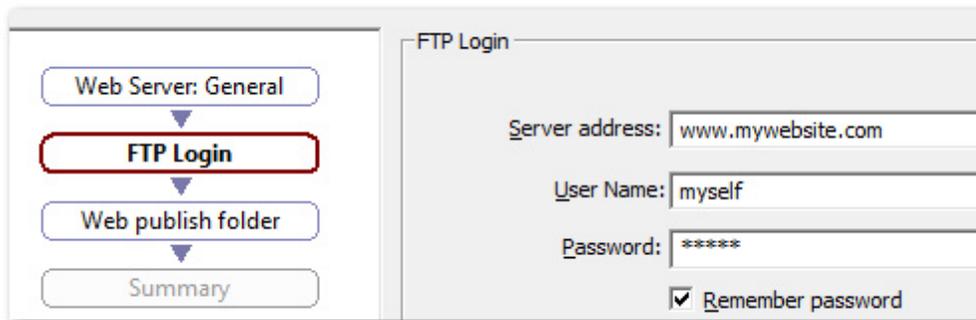
Let's see how we can configure this with Simfatic Forms.

Configuring the server in the 'Add Server Wizard'

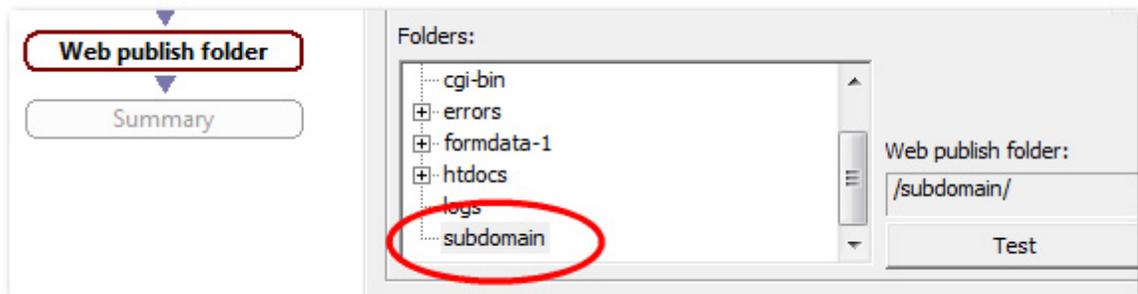
- Open Simfatic Forms and create the form
- Go to 'Take the code' page.
- Press the 'Take the code' button to start the wizard.
- Choose 'Simfatic Forms Swift Upload' in the 'upload method' page. Press 'Next'
- Press the 'Add Server' button in the Swift Upload page
- Provide the sub-domain address in the 'Webserver:General' page.



- Update the FTP login:



- Press the Next button. In the 'Web publish folder' page, select the folder corresponding to the sub-domain. (folder /subdomain/ in this sample case)



- Finish the 'Add Server Wizard' and press the 'Upload' button to upload the form processor scripts.

3.16 How to add custom functionality to your form

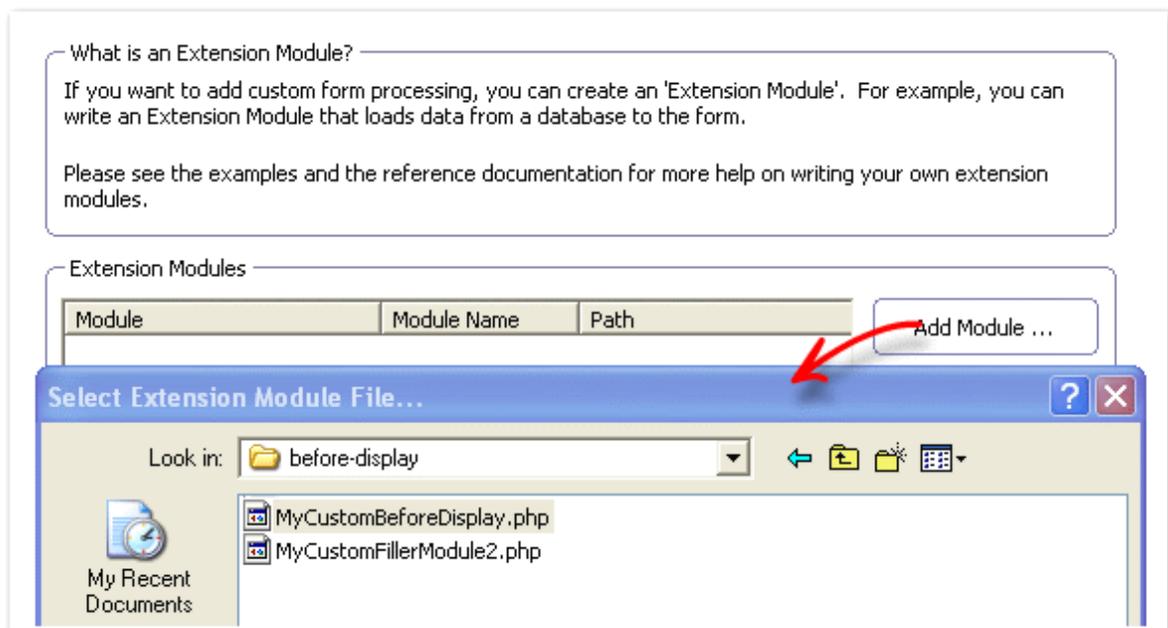
In addition to the default processing provided by Simfatic Forms (like sending email, saving to file/database etc) you can add your own custom processing to the form.

You can write an 'Extension module' to add custom functionality to your form.

Extension modules can be written to pre-load the form, do custom form validations or to handle the data submitted in the form.

These are the steps to add a custom Extension Module to your form.

1. First you have to write the Extension module in PHP and save to a file. (Refer the Extension Modules reference page for more info.)
2. Enable Extension module in the 'Form Processing Options' page step.
3. Add your extension module in the 'Extension Modules' page



See the Extension Modules reference page for more information on Extension

Modules.

If you only want to modify the client side code of the form (like the form display style or Javascript customizations) you can use the Custom code feature.

See Also:

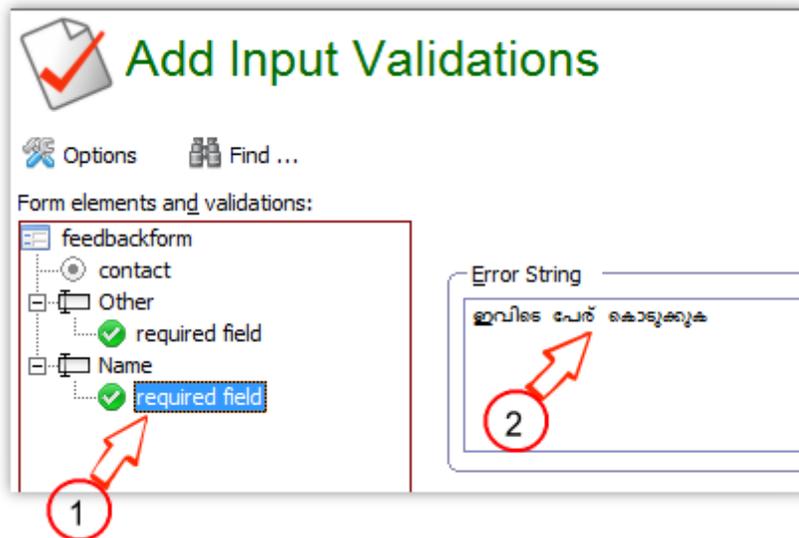
[Extension modules function reference](#)

[How to pass a value to the form](#)

[How to handle JavaScript events in the form](#)

3.17 How to edit the validation error messages

If you want to edit or translate the default validation error message just follow these steps:

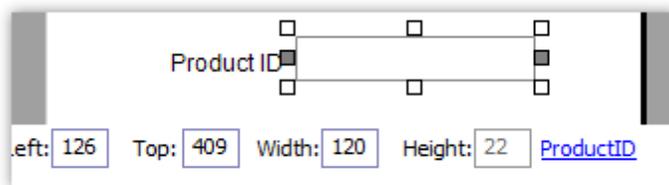


- select the validation node (with the green tick) on the left side
- edit the error message on the right side

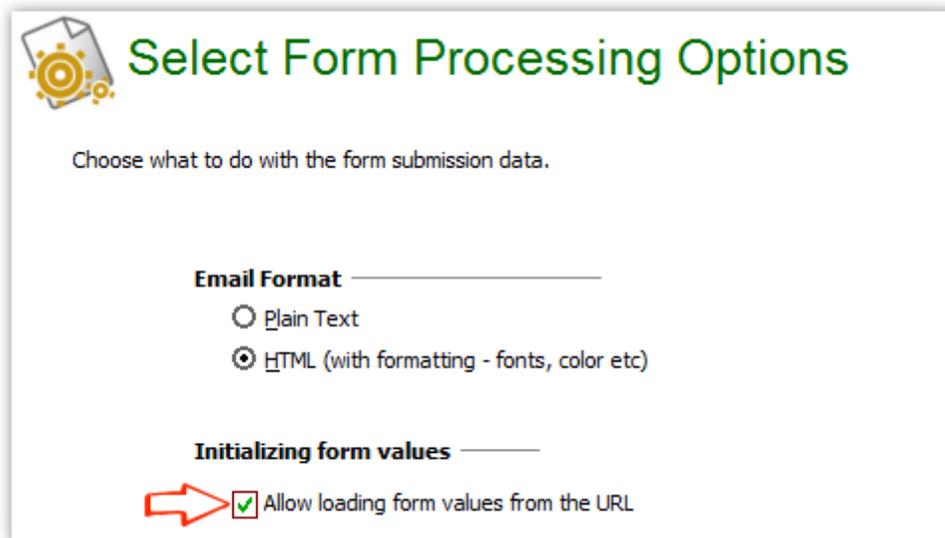
3.18 How to pass a value to the form

If you want to pass a custom value to the form (like the referring ProductID), follow these steps:

Add the field on your form and name it (for example ProductID).

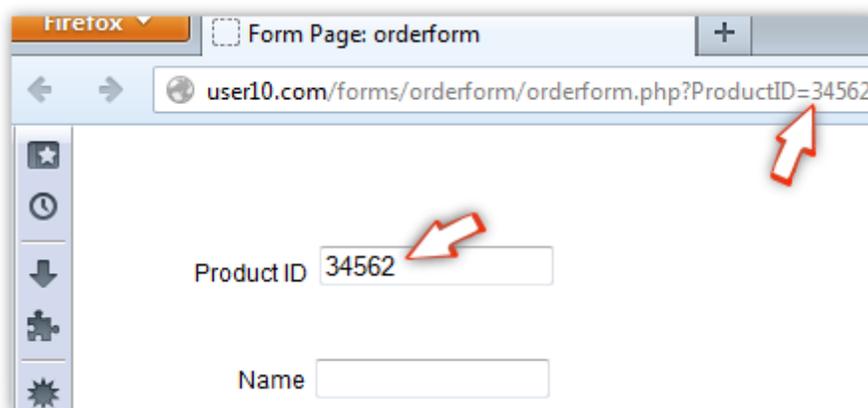


In the 'Form processing options' page -> 'General settings' section, enable 'Allow loading values from URL' option.



Upload your form.

Now you can pass the value through the URL.



 When there are more than one values to be passed through the URL, separate with **&**

Example:

```
http://user10.com/forms/orderform/orderform.php?  
ProductID=34562&Name=John&Email=jj@domain.com
```

 The field names are **case sensitive**. ProductId=34562 will not work in the example above.

 You can use a hidden field for tracking the value as well

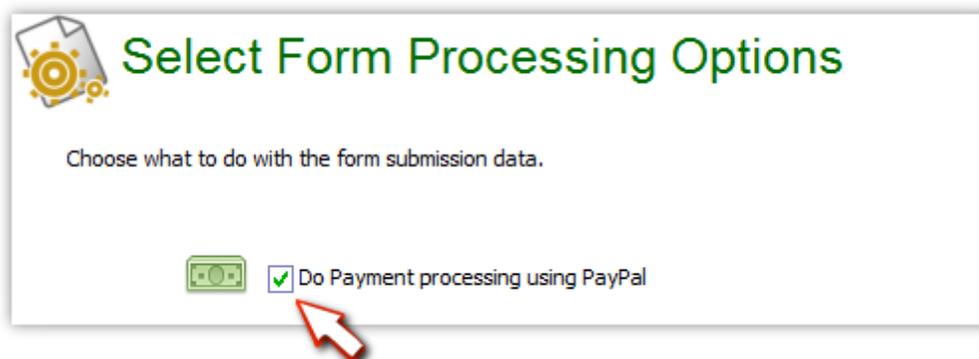
 If you are embedding the form using iframe, add the field values to the iframe's URL

```
<iframe  
  src='orderform.php?sfm_from_iframe=1&ProductID=3456'
```

3.19 How to accept payments through Paypal

Simfatic Forms supports Paypal website payments standard. Your user will be redirected to the Paypal website, where she can make the payment and come back to your website.

- First, signup with Paypal.
- In Simfatic Forms->Form processing options page enable 'payment processing using Paypal' option



Press Next and proceed to the 'Paypal setup' page.

- In the Paypal setup page, press the 'Configure Paypal' button.
- In the Paypal setup wizard, provide the email address you signed up with Paypal.
- Choose the payment type (Products, subscription or donation) and press Next.

In the Product/Subscription setup pages, you can select fields from the form. For example, if you have a calculation field in your form to calculate the total payment to

be made, you can just select that field name in the product setup page.

Setup : Setup Product/Service

General Settings
▼
Products
▼
Additional Charges
▼
Summary

Item Name: ▶

Unit Price: ▶

Quantity: ▶

You can have a fixed value, or a value from the form. Press the '>>' button to select a form field.

+ Add more items

- Press Next and complete the pay pal setup. Upload your form.

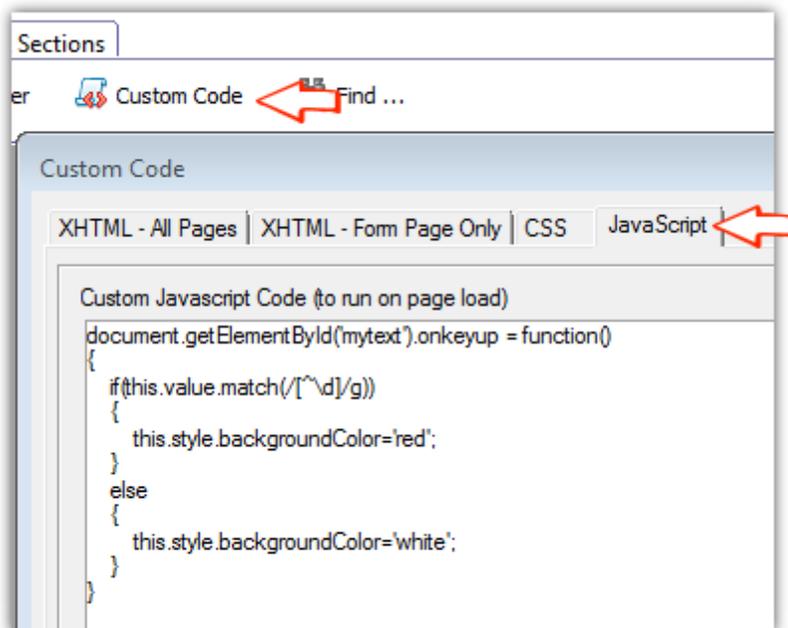
 It is a good idea to get a paypal sandbox account (<https://developer.paypal.com/>) and test the Paypal setup first.

3.20 How to handle JavaScript events in the form

If you want to add some custom event handling code, you can use the custom code feature of Simfatic Forms.

Here is one simple example. The following Javascript code changes the background color if a non digit character is typed in to the input field:

```
document.getElementById('mytext').onkeyup = function ()
{
    if(this.value.match(/[^\\d]/g))
    {
        this.style.backgroundColor='red';
    }
    else
    {
        this.style.backgroundColor='white';
    }
}
```



Using jQuery

Simfatic Forms supports jQuery, a powerful Javascript library. This page shows you how to handle events unobtrusively using jQuery.

Imagine you want to change the background color of a text box on setting the focus, follow these steps:

First, find the ID of the input element: select the element in the 'draw the form' page. The element's ID is displayed in the bottom



The following snippet of code changes the background color on focus change.

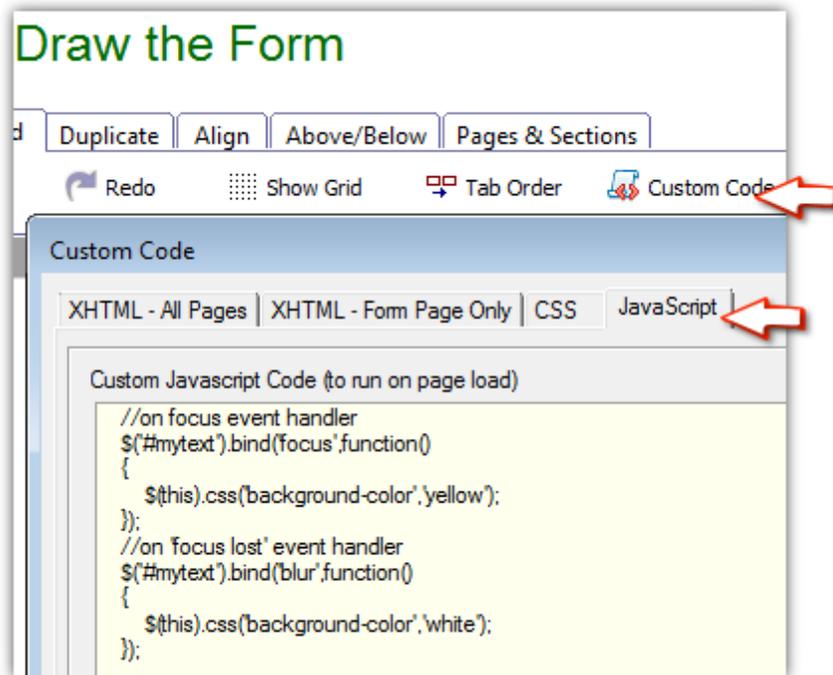
```
//on focus event handler
$('#mytext').bind('focus', function()
{
    $(this).css('background-color', 'yellow');
});
//on 'focus lost' event handler
$('#mytext').bind('blur', function()
{
    $(this).css('background-color', 'white');
});
```

The 'focus' event handler changes the background color to yellow. On focus loss, we change the background color to white.



Notice how the ID of the element is used to handle the events only for *that* element

Now, this custom code can be put in the Draw the form->Custom code box

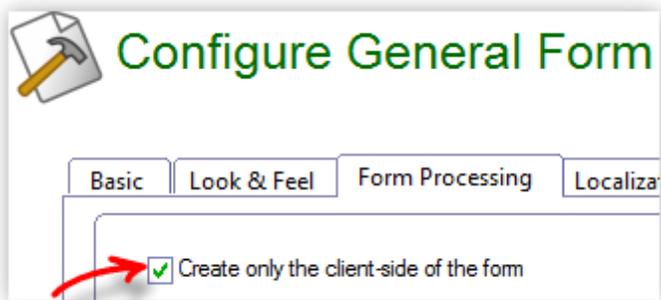


Similarly, you can handle any type of events (onchange, keyup) on any element or group of elements in your form.

 Use Firebug(getfirebug.com) with Firefox browser to debug your custom code

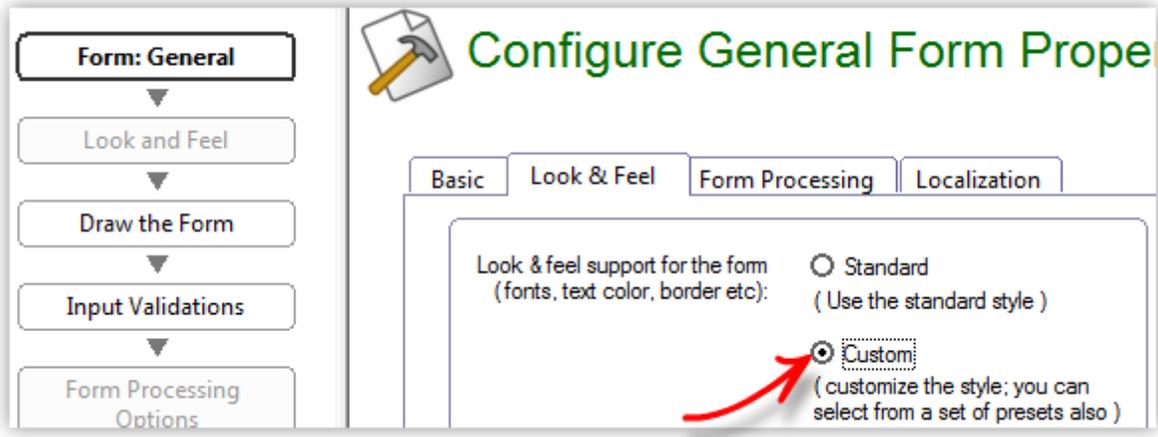


Switch to 'client side only' mode temporarily to test the custom code locally.
(The option is in Form:General page, 'Form processing' tab)

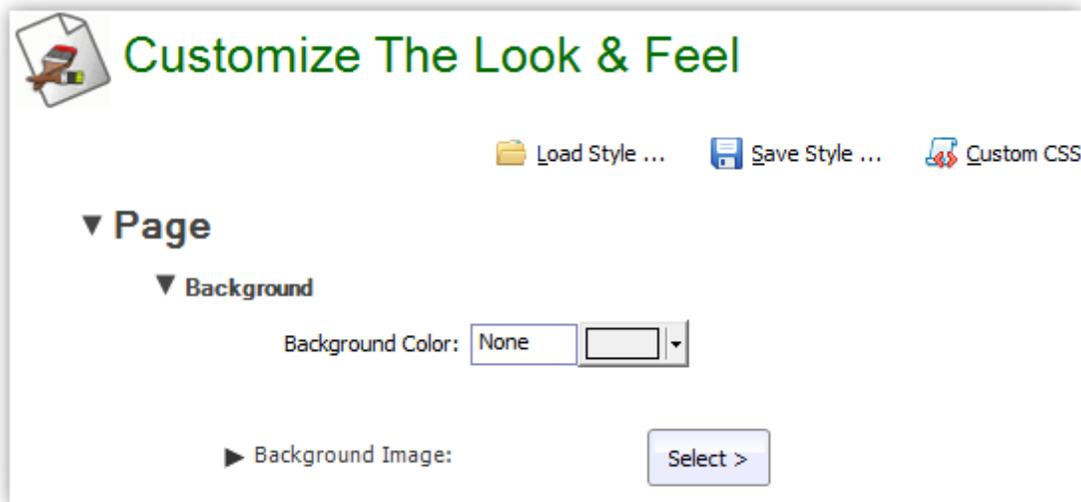


3.21 How to customize the Look & Feel of the form

In the Form:General page, Look & Feel tab select the 'custom' option



Press 'Next' and in the Look and Feel Page you can edit and customize the style of the form



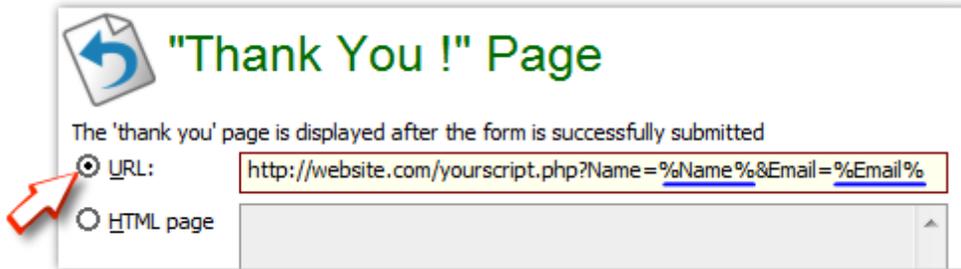
You can load pre-built styles pressing the Load Style button. Similarly, you can save your custom style and apply it to your other forms.

 You can use the custom CSS button to further customize the style using CSS code.

3.22 How to redirect form submission values to another script

If you have another custom script that you would like to trigger after the form processing by Simfatic Forms is done, you can follow these steps.

In Simfatic Forms-> 'Thank you' page , provide the URL of your custom script.



 **"Thank You !" Page**

The 'thank you' page is displayed after the form is successfully submitted

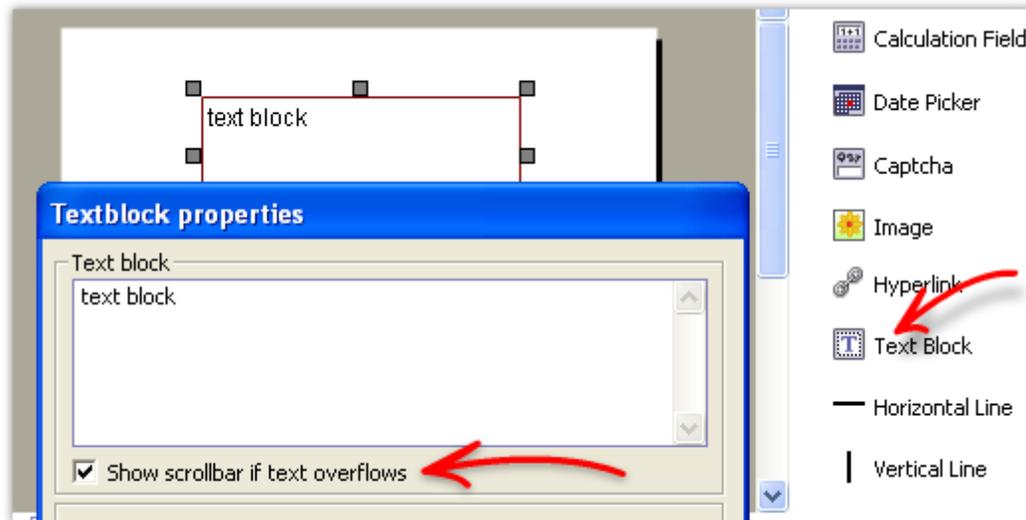
URL:

HTML page

Please note how the form submission values (Name and Email) are passed to the custom script

3.23 How to make a Terms & Conditions box

To make a scrollable block of large amount of text, you can use the text block. This is often used to show a terms & conditions box.



Put a textblock on your form, enter the block of text in the text block.

Then enable the 'Show scrollbar' option. Terms & conditions used to be very lengthy text that you can put in the scrollable text block.

3.24 How to seamlessly integrate with your website

After you have uploaded your form to your website, you can embed the form in any of your web pages.

You can create a new page (just like any web page on your website) with header, footer and all common elements.

From Simfatic Forms->Take The code->Copy Form code copy the embed the form code and paste it.

The Form can be made a popup as well.

Add a link to your website to popup the form. Get the popup code from 'Copy Form code' box and update the link.

3.25 How to Integrate the form

In the 'Take the code' page, follow the wizard steps to upload your form to your website.

Once the 'Take the code' steps are completed, you will get the 'Copy form code' option.

Press the 'Copy form code' button. You will get different options to integrate the form with your website.

You can embed the form in your web page or can add a link to open a popup with the form.

The pages below gives more detailed steps on embedding the form with common HTML editors and CMS

- Dreamweaver
- Expression Web
- Joomla
- Drupal

Even if your favorite HTML editor is not listed here, you can just follow these general steps:

For HTML editors:

- Open your page where you want to embed the form
- Switch to HTML source mode
- Copy the Embed using iframe' code from Simfatic Forms and paste in your HTML page

For CMSes

- Make sure your page editor does not filter out iframe code.
- Switch to HTML source mode
- Copy the Embed using iframe' code from Simfatic Forms and paste in your page

Dreamweaver



Note: You have to upload the form to your web site first (why ?)

- Open Dreamweaver and open the page where you want to embed the form
- Place some text "My Form Here" at the location where you want to embed the form. Select the text.
- Select menu Item View->Code
- In Simfatic Forms->"Take the code" page, press the copy form code button.
- Copy the 'Embed using iframe' code
- Back in Dreamweaver replace "My Form Here" with the copied form code
- Save & upload the page

Expression Web



Note: You have to upload the form to your web site first (why ?)

- Open Expression Web and open the page where you want to embed the form
- Place some text "My Form Here" at the location where you want to embed the form. Select the text.
- Click the 'code' tab at the bottom of the editor. The editor switches to HTML source editor.
- In Simfatic Forms->"Take the code" page, press the 'copy form code' button.
- Copy the 'Embed using iframe' code
- Back in Expression Web, replace "My Form Here" with the copied form code
- Save & upload the page

Joomla



Note: You have to upload the form to your web site first (why ?)

The Tiny MCE editor in Joomla will not allow iframe tag by default. So first, we have to get iframe to the list of allowed tags

- In Joomla, go to Extensions -> Plugins -> TinyMCE
- In "Extended Valid Elements", enter the following:

```
iframe[src|style|width|height|scrolling|marginwidth|  
marginheight|frameborder|allowtransparency]
```

- Go to the Joomla page where you want to add the form and edit the page
- On the editor text area click on the HTML button and switch to HTML edit mode
- In Simfatic Forms->"Take the code" page, press the 'copy form code' button.
- Copy the 'Embed using iframe' code
- Back in Joomla, paste the code in the editor.

Drupal



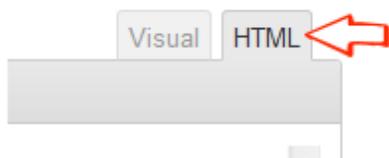
Note: You have to upload the form to your web site first (why ?)

- Go to the page where you wan to embed the form
- Edit the page and in the Input Format section select "Full HTML"
- In Simfatic Forms->"Take the code" page, press the 'copy form code' button.
- Copy the 'Embed using iframe' code
- Back in Drupal, paste the form code.

WordPress

 Note: You have to upload the form to your web site first (why ?)

- Go to the page or post where you wan to embed the form
- In the Post area, select HTML editing mode.



- In Simfatic Forms->"Take the code" page, press the 'copy form code' button.
- Copy the 'Javascript embed the form' code
- Back in Wordpress, paste the form code.

3.26 How to make a popup form

Suppose you want to popup the form on clicking a link or a button in your website. Here is how to make such a pop up form.

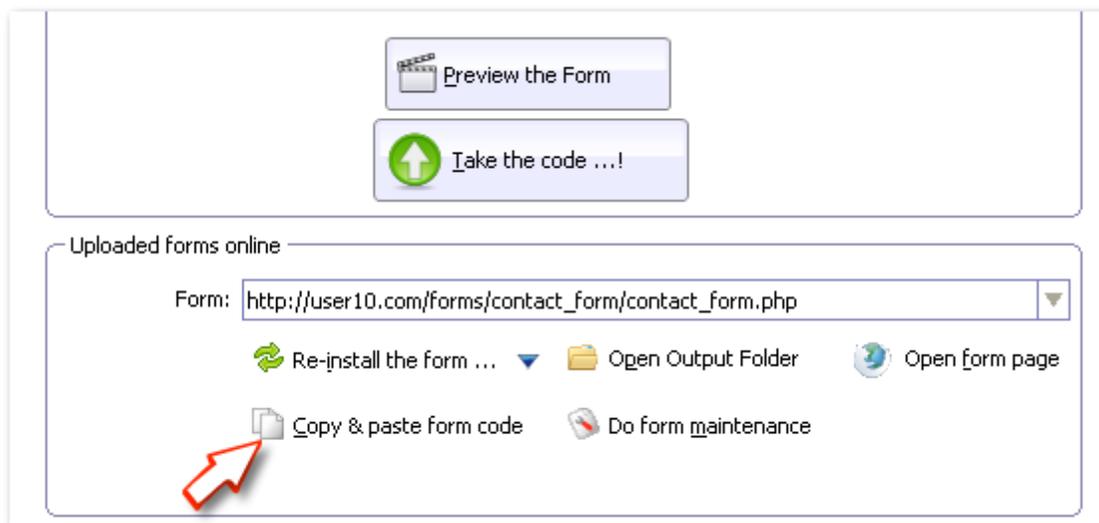


Popup Form

Create the form in Simfatic and follow the "Take the code" steps to upload the form to your website.

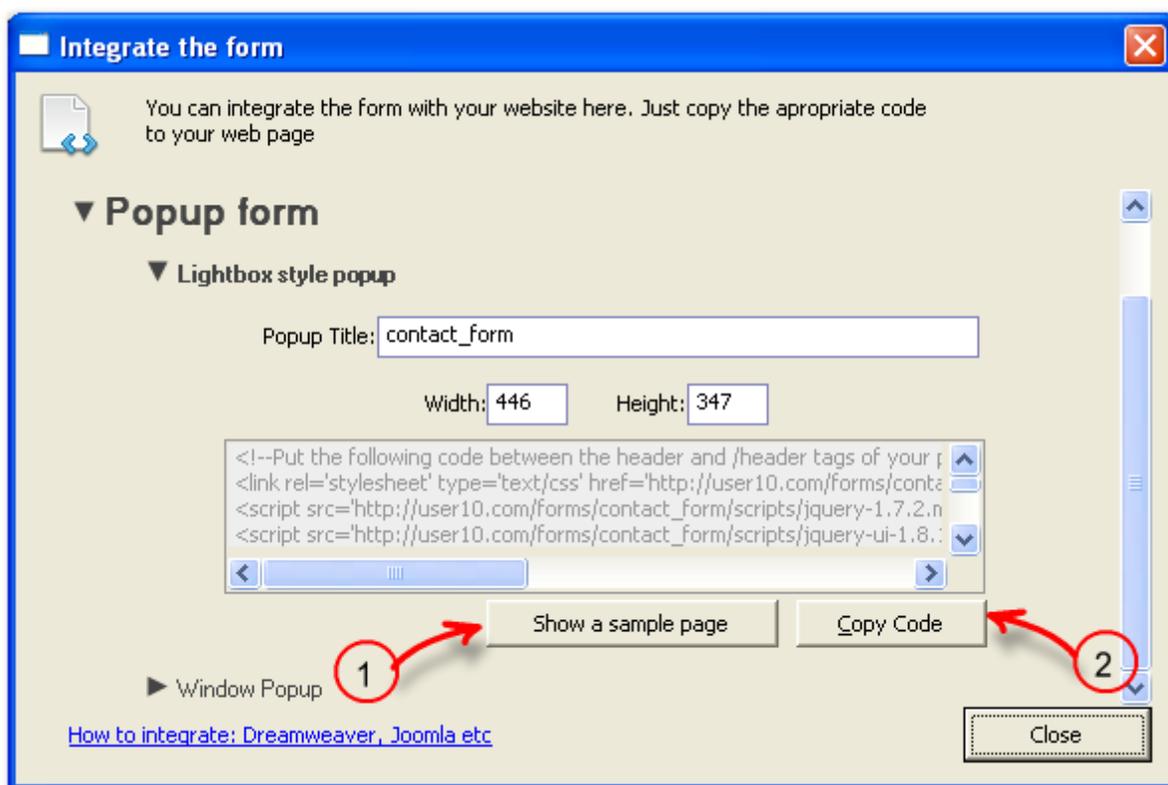
(You have to first upload your form to your website. why ?)

After uploading the form, you will get a copy form code button in the "take the code" page.



Press the 'copy form code' button

Under popup form->lightbox style popup press the "show a sample page" button to see a sample page opened in your browser with a popup form link.



Press the copy code button

```
<p>
    <!--The following code is the Hyperlink for the form. You
    can put it where you want the popup link.-->
<a class='sfm_contact_form_link' href='javascript:void(0);'>Open the
form contact_form</a>
</p>
<p>
    Ut wisi enim ad minim veniam, quis nostrud exerci
tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo
```

Copy the code to your web pages

Press the copy code button and paste the code in your web pages using your HTML

editor.

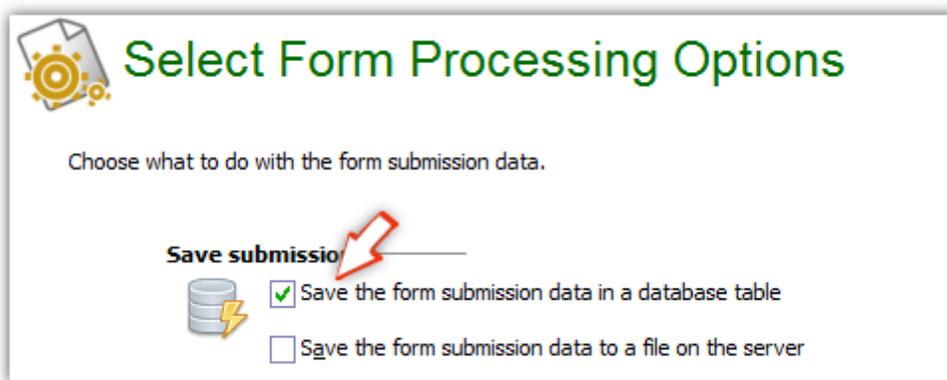
The code contains instructions where to put the code. (like the <head> section of the HTML page).

You can further enhance the link (for example, using an image instead of text) using your HTML editor.

3.27 How to view form submissions online

You can view the form submissions online in the Form Admin Page. To view the form submissions online, you have to enable saving to database.

If you already haven't already enabled Save form submissions to database option, enable it in the "Form processing options" page.

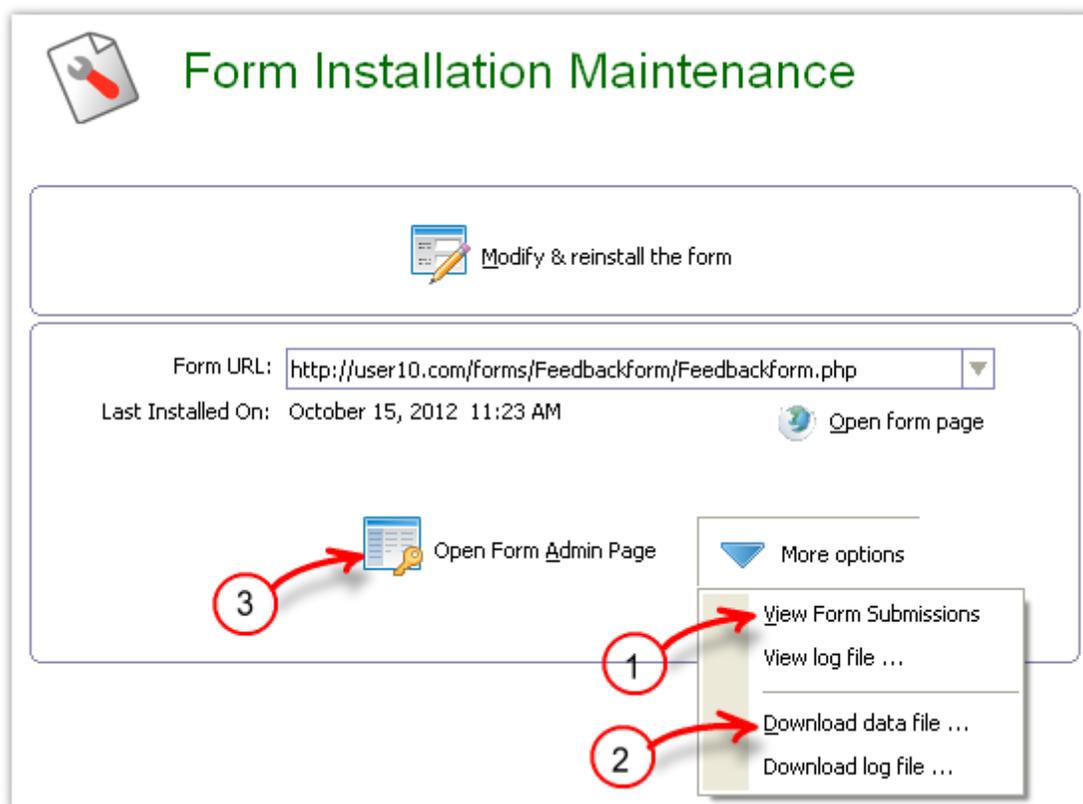


Then press "Next" and proceed to re-upload the form.

Go to the form maintenance page by pressing this icon

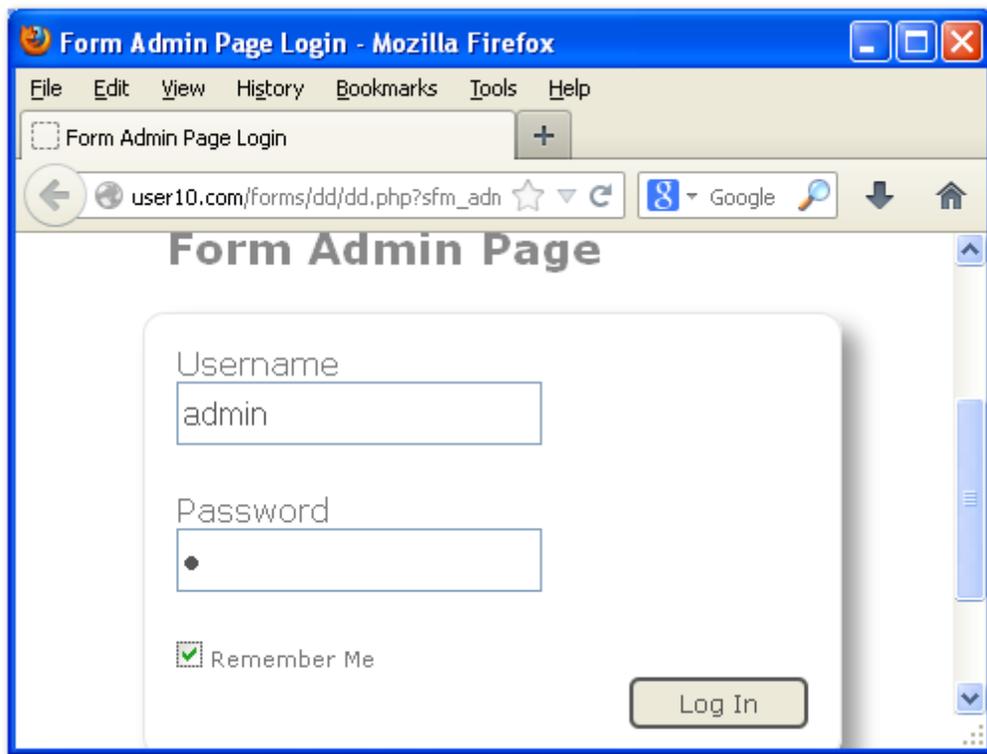


In the maintenance page, there is a "Form Admin page" button. Press the button to open the Form Admin page in the browser.



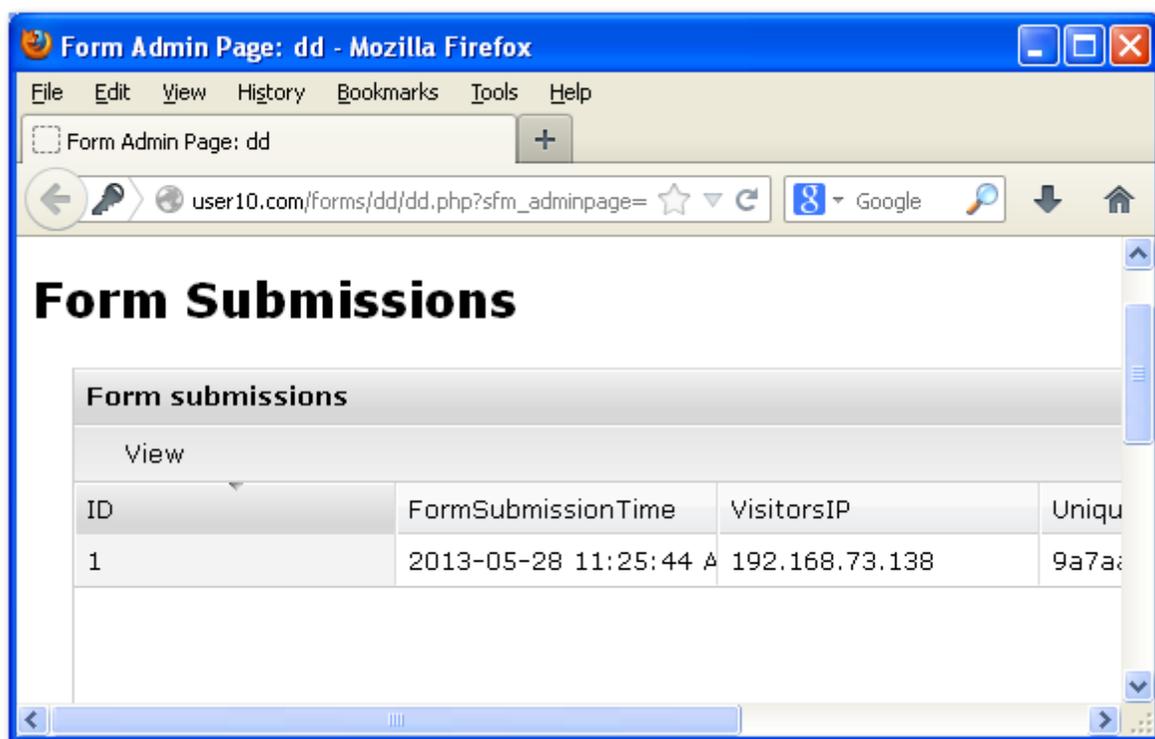
Form Admin Page Button (option 3) in the Maintenance page

In the browser, it will open the form admin login page

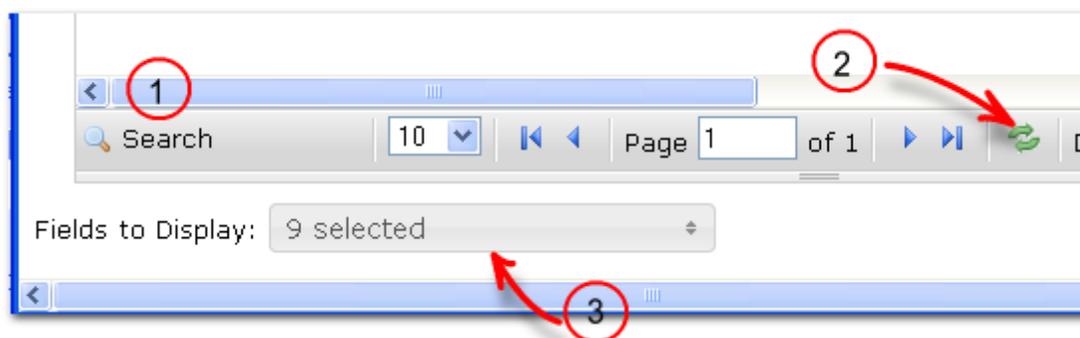


Enter the same login you created while uploading (in the "Take the code" ->Admin Login Step)

The First section of the form admin page displays the form submissions.



There are several options in the bottom of the submissions table:



1. Search button 2. Refresh 3. Select the fields to display

See Also:

Form Admin Page

Part

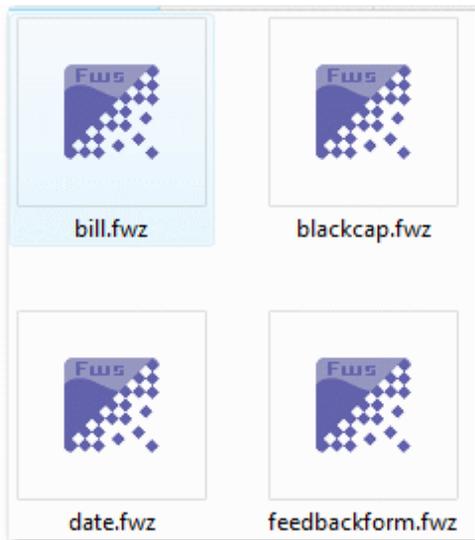


Information Reference

This section contain general reference information.

4.1 The Simfatic Forms project file

The Simfatic Forms project file keeps your form design and all the relevant information (example: The web servers where you installed the form).



The file has extension fwz. While making the form in Simfatic Forms, you can save the form design by pressing the Save button in the toolbar (or by choosing the File->Save menu item). Later you can continue working on the form by opening the same Project file.

Never upload the fwz file to your web server!

The fwz file (the Simfatic Forms project file) is a proprietary file format that makes sense only to Simfatic Forms application. The scripts generated by Simfatic Forms can't read the fwz file. So, there is no need of uploading the fwz file to the web server.

The fwz contains security information regarding your form (like a form id and

signature that is required to download form submission data). Therefore, it is potentially unsafe to upload the fwz file to your web server.

Keep the fwz files of your forms in a safe location. Make sure that the fwz files are included in your regular backups.

4.2 Naming Rules

There are certain restrictions by which you can create a name (name of the form and that of the elements)

- i. The name can contain alpha-numeric characters and _ (underscore) only. No space or other punctuation characters are allowed in the name.

- ii. The name should start with an alphabetic character.

Wrong

Contact Form

(space in name)

My-Contact-Form

(hyphen '-' in name)

1Name

(starts with number)

Right

ContactForm

My_Contact_Form

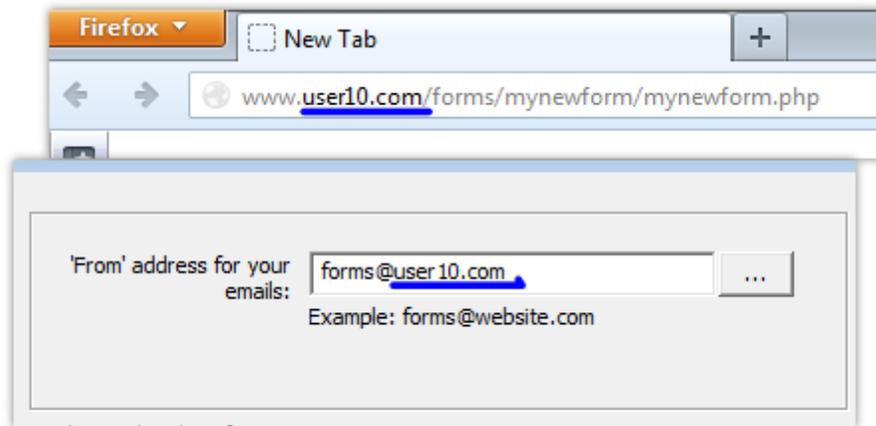
signupform

signupform1

4.3 Enter correct 'From' address for proper Email delivery

The 'From' address used for sending emails should belong to the same domain as your script. Else, most servers would reject your email .

For example:



When you are sending an email with From: someone@user10.com, the email servers will verify the domain records of user10.com that your server is *authorized* to send email on its behalf.

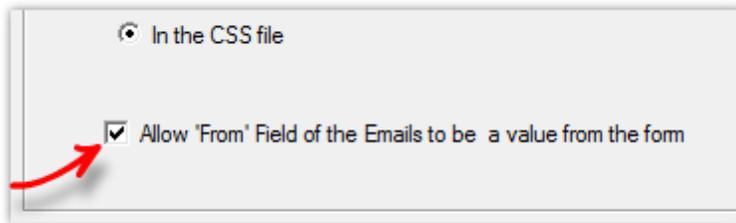
It is possible to setup SPF record in the domain authorizing another domain to send email on its behalf. (See SPF validation)

Simply put, it is safer to use user@domain.com as the 'From' address when your script is running on domain.com.

This is why the form submission emails can not have the user's email address in the 'From' field. Simfatic Forms will put the user's email address in the 'Reply-To' field and you can quickly reply to the form submission email.

Forcing Simfatic Forms to put the user's email in the 'From' field

In Simfatic Forms menu item Tools->Options ->Code generation tab, select 'Allow From field to be values from the form'



After making this change, re-upload the form.

4.4 Calculation Field

Simfatic Forms has a powerful feature to create online calculators. The 'Calculation field' can do calculations 'on the fly'.

You can provide a formula for the calculation field. The formula can take fields from the form as 'variables'. In addition to all common arithmetic, mathematical operations, a number of in-built functions are also provided.

Creating a calculation field

First, place all the input elements that will be part of the formula. Rename the input elements if required. You will mostly need 'Decimal Number' fields; but you can use other input elements as well (like a general text field, radio button, or a drop down list)

Now place a Calculation field on the form. Double-click the calculation field to open the properties of the Field.

Press the 'Edit Formula' button to compose the formula.

The Formula editor contains three parts: a 'Form Fields' box that lists the elements in the form, a keypad for digits & operators and a box that lists all the functions supported in the formula.

Creating complex calculation fields

If the calculation is very complex, it will be handy to break the calculation in to several steps.

You can create several calculation fields each calculating a part of the formula. It is possible to make these partial results hidden if required. (check the 'Hidden' check box in the Calculation field's property box).

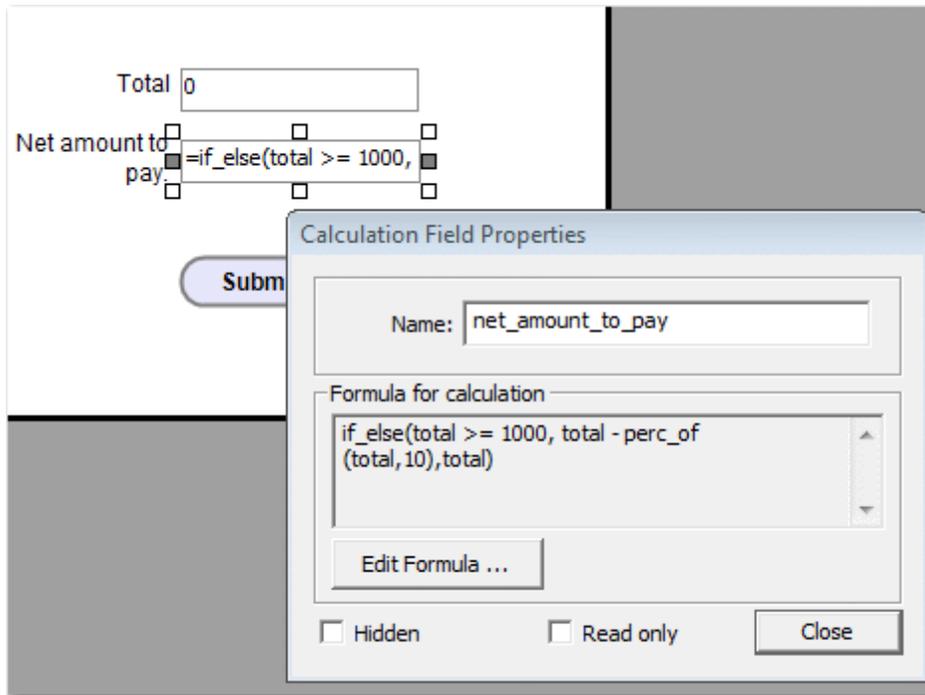
Using conditions in the formula

It is possible to use conditions in the calculation field's formula. For example, imagine you want to give 10% discount if the total purchase is above or equal to 1000 dollars.

You can use the function `if_else()`. The `if_else` function has three arguments: the condition, 'true part' that will be returned if the condition is true, and 'false part' that will be returned if the condition is false.

We can create a calculation field for the `net_amount_to_pay` having formula:

```
if_else(total >= 1000, total - perc_of(total,10),total)
```



perc_of is another handy function to calculate the percentage of a value.
 perc_of(total,p) calculates p percentage of 'total'

Using radio buttons or a dropdown list in the formula

It may be required some times to base the calculation on a selection (radio button or drop down list)

Ensure that you have provided 'value' for all options in the radio group. If you use the radio group field in the formula, the value of the chosen item will be used. For example, imagine you have different subscription plans each costing 10, 20 or 30 ('base' group) and 1 5 and 6 ('Add On Group').

The plans are provided as two sets of radio buttons ('Base' and 'AddOns') You can have the 'value' for each option same as the price.

The Calculation field for the Total price can have the formula as:
base + AddOn

Since it will take the value of the selection on calculation, the right prices get added to the total.

See Also:

How to create an online calculator

Calculation field : function reference

Formula editor

Calculation field properties

Calculation field : function reference

There are several functions that you can use in a calculation field. The functions can be categorized as Mathematical functions, Conditions, and date functions.

Note that the arguments to these functions can be fields in the form or constants.

General functions

if_else(condn, true_part, false_part)

Evaluates the condition 'condn' and returns 'true_part' or 'false_part' depending on the evaluation result of the condition.

Examples:

```
if_else(total>1000,10,0)
```

if the value entered in the field total is greater than 1000, return 10.

If total is less than or equal to 1000, return 0.

is_empty(input_name)

Checks whether the input field is empty or not. Returns 1 if empty 0 if there is value in the field. If the input is only spaces, it is taken as empty.

map_to_number(mapstr,value)

map from string value to number. This function is useful for including radio groups or list boxes in a formula.

The format of the string value map is "string1 => value1, string2=>value2, string3 => value3 " and so on.

Example:

imagine you have a 'subscription' radio option with three cases: none, basic and advanced with cost 0, 100 and 200.

The following call can be used in this case:

```
map_to_number("none=>0, basic=>100, advanced=>200",  
subscription)
```

Date Functions

days_between(date1,date2)

returns the number of days between two dates (result = date1 - date2)

The return value can be negative if date1 is before date2. date1 and date2 should be DateFields.

days_from_today(other_date)

returns the number of days from today to other_date. (that is, result = other_date - today) The return value can be negative if other_date is in the past.

age(date)

returns the number of years after date. This function can be used to determine the age if you have a date of birth field in your form. If the date is in the future, the return value will be negative.

Deprecated Functions

These functions are for backward compatibility. There are alternatives for these functions in the current version.

format_currency (amount, add_comma)

The decimal places are rounded to two digits. (example: 12.129 becomes 12.13)

If there is only one digit after the decimal point, 0 is appended.(example:12.5 becomes 12.50)

The currency format depends on the localization settings selected in the Form:General page->Localization tab



In the current version, you can enable 'Format as currency' in the calculation field properties box instead of using this function.

Examples:

```
format_currency(12,0)
```

\$12.00

```
format_currency(12.1,0)
```

\$12.10

```
format_currency(12.129,0)
```

\$12.13

```
format_currency(1000,0)
```

\$1,000.00

is_checked(checkbox_name)

Useful for including a checkbox in a formula. Returns 1 if the check box is selected, 0 otherwise.

Example:

Suppose subscribing to the newsletter enables a 10% discount. You can do this using formula:

```
is_checked(checkBoxSubscribe) * 10
```



In the current version, you can directly use the check box in the calculation formula

```
example chkSubscribe * price
```

is_checked_grp(value,checkbox_group)

This function can be used for including a check box group in a formula. It checks whether the check box with the given value is selected. Remember that one can select multiple options from a check box group. Returns 1 if selected, 0 otherwise.

Example:

```
is_checked_grp("PCMag",Magazines)
```



In the current version, you can directly use the check box in the calculation formula

```
example Magazines["PCMag"] * price
```

Mathematical functions

The mathematical functions are for doing general mathematical operations.

max(p1,p2,p3,...)

find the maximum value from the arguments. You can pass any number of arguments.

min(p1,p2,p3,...)

find the minimum from the arguments. You can pass any number of arguments.

avg(p1,p2,p3,...)

find the average of the passed-in values. You can pass any number of arguments.

abs(num)

Returns the absolute value of the number

Examples:

abs(-3) is 3

abs(3) is 3

abs(-3.44) is 3.44

ceil(num)

returns the number rounded upwards to the nearest integer

Example:

ceil(2.1) is 3

ceil(2.001) is 3

ceil(-2.001) is -2

floor(num)

returns the value of the number rounded downwards to the nearest integer

Examples:

```
floor(2.1) is 2
```

```
floor(2.001) is 2
```

```
floor(-2.001) is -3
```

round(num,dec)

rounds the number to dec decimal places.

num: the number that should be rounded.

dec: the number of decimal places to round

Examples:

```
round(2.1115,2) is 2.11
```

```
round(2.1115,3) is 2.112
```

```
round(2.5,2) is 2.5
```

```
round(2.5,0) is 3
```

pow(x,y)

returns the value of x to the power of y.

Examples:

```
pow(3,2) is 9
```

`pow(3,3) is 27`

sqrt(x)

calculates the square root of x

log(x)

calculates the natural logarithm(Base E) of x

perc_of(total,perc)

returns the 'perc' percentage of total. For example, to find the 10% of amount, the formula is `perc_of(amount,10)`

Example:

`perc_of(1000,10) is 100`

find_perc(fraction,max)

calculate the percentage value of fraction, given max.

for example: `find_perc(100,1000) is 10`

Trigonometric Functions

The following trigonometric functions also are supported

`sin(x)`

`cos(x)`

`tan(x)`

$\text{asin}(x)$

$\text{acos}(x)$

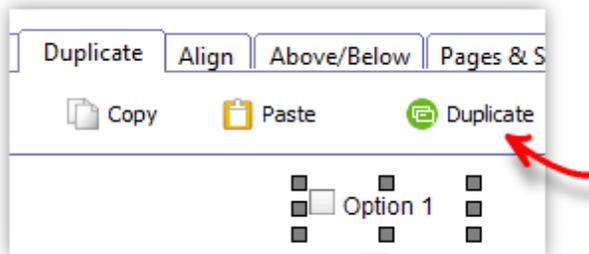
$\text{atan}(x)$

4.5 Using the duplicate feature

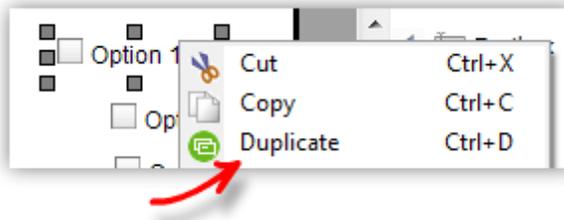
You can use the duplicate button to create copies of an element quickly.

This is especially useful while creating radio buttons and check box groups.

Select the element and press the duplicate button



Alternatively, you can select the element and choose duplicate from the context menu



You can use the keyboard shortcut (Ctrl + D) as well.

4.6 Is Captcha required in your form?

Captcha is used in forms to make sure that the form is submitted by a user.

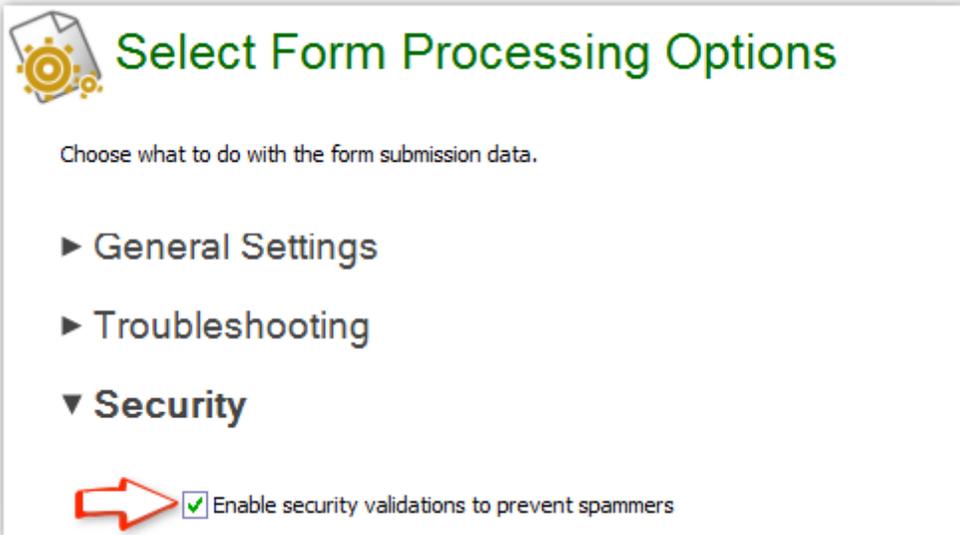
There are software tools that can 'read' your web page, parse your form and automatically submit the form just like a human user would. Usually, such 'form submission bots' post spam advertisements, expecting clicks from you.

Captcha prevents such submissions by displaying an image that can be read only by a human user. So the Captcha images are difficult to read. There are random lines and dots, characters are turned and skewed so that the 'automated bots' can not read it.

Captcha **does not** add any other security to your form or your website.

Adding Captcha to your form considerably increases the difficulty for your user to submit your form. A very good percent of your web page visitors will abandon the form itself fearing the Captcha.

Simfatic Forms has security features to prevent automated submissions **without using Captcha.**



The image shows a dialog box titled "Select Form Processing Options" with a gear icon. Below the title, it says "Choose what to do with the form submission data." There are three expandable sections: "General Settings", "Troubleshooting", and "Security". The "Security" section is expanded, showing a red arrow pointing to a checked checkbox labeled "Enable security validations to prevent spammers".

Select Form Processing Options

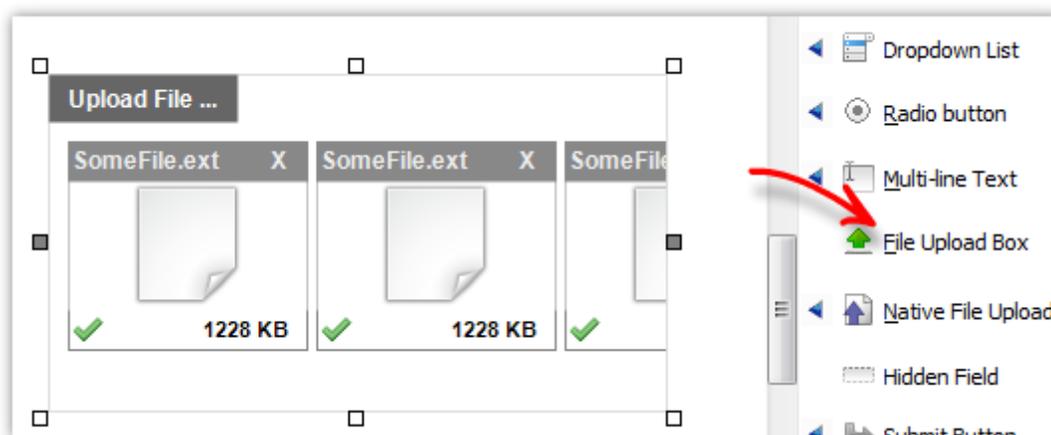
Choose what to do with the form submission data.

- ▶ General Settings
- ▶ Troubleshooting
- ▼ **Security**
 - Enable security validations to prevent spammers

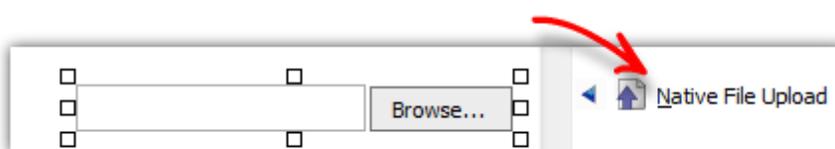
Save your users from trouble; do not add captcha to your form.

4.7 Prefer File uploadbox rather than Native file upload

The file upload box offers better features compared to the native file upload.



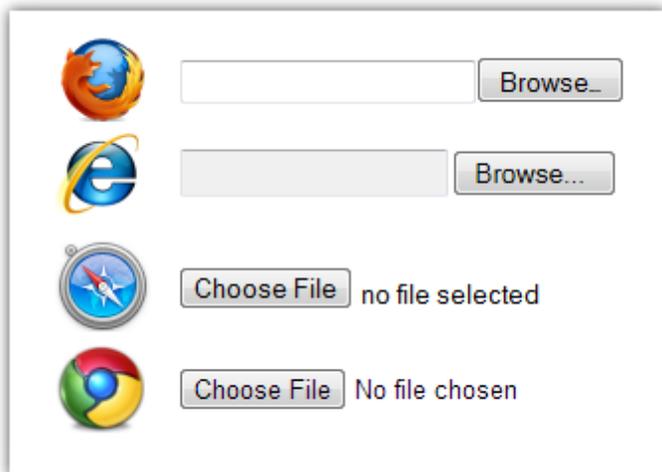
file upload box



Native file upload

- The file upload box shows a preview of the file uploaded (for image file types)
- You can upload more than one files in single file upload box
- When you have multiple pages in the form, the file upload box is more useful. The file upload box retains the files uploaded as the user traverses back and forth between pages.

- Similarly, if you have enabled 'confirm page' step, the file uploadbox retains the uploaded files if you press edit in the confirmation page.
- Different browsers display the native file upload box in different ways.



4.8 Smart Validations

The key features of smart validation are:

- The validation is triggered based on the validation type.

For example, 'maximum length' validation is triggered as the user types in the input. email validation is triggered when the user switches the focus.

This allows earliest and 'live' feedback and correcting the errors earlier, rather than in one shot on form submission.

- Once there is an error in a field, and as the user corrects the error, the field is validated again.

The error message disappears automatically when the field is valid.

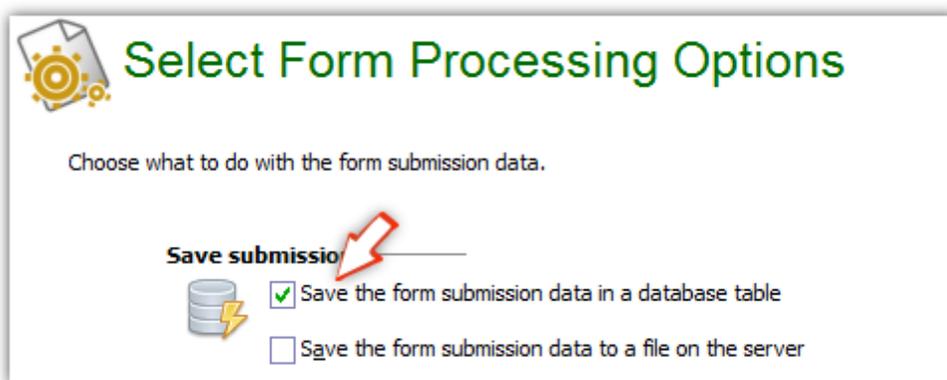


If you prefer one shot validation on submit, you can select that option in the 'Input validations' page->'Validation options'

4.9 Unique validation requires Database

The Unique validation requires database support to do the validation. When the form user enters a value, the validation has to check whether that value was entered before (hence making sure the value is *unique*). When the form submissions are saved to a database table, the validation can just check the database table.

Therefore, the 'Save to database' option in the 'form processing options' page should be enabled for the 'unique' validation to work.



Select Form Processing Options

Choose what to do with the form submission data.

Save submission

Save the form submission data in a database table

Save the form submission data to a file on the server

4.10 Extension Modules

Extension modules can be used to add your own form processing features to your form.

Examples:

- You want to load data from a database to the form
- You want to save the form submission data to a more complex schema of tables in a database.
- You want to submit the form submission to a web service

and so on.

The structure of an extension module

An extension module is written in PHP. You should be familiar with the syntax of PHP to write extension modules.

Your extension module should extend `FM_ExtensionModule` class.

You can override the functions of `FM_ExtensionModule` and do custom handling. The functions of the extension module are called at specific points of form processing. For example, the function `BeforeFormDisplay()` is called just before displaying the form; the function `FormSubmitted()` is called after the form is submitted and data is validated.

The extension module below loads the form (the field 'Name')

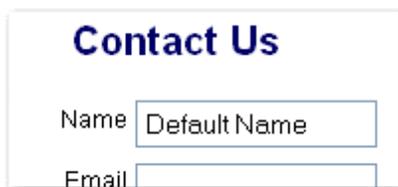
```
<?PHP
```

```
class MyCustomLoadingModule extends FM_ExtensionModule
{
    function BeforeFormDisplay(&$formvars)
    {
        $formvars["Name"] = "Default Name";
        return true;
    }
}

?>
```

The code above loads a field 'Name' with a default value. To see this code in action, put the code above in a file (say mymodule.php) Then Enable 'extension modules' in the 'Form processing options' page, and add the file in the 'Extension Modules' page.

The field "Name" will have the value "Default Name" when the form is loaded.



The image shows a screenshot of a web form titled "Contact Us" in bold blue text. Below the title, there are two input fields. The first field is labeled "Name" and contains the text "Default Name". The second field is labeled "Email" and is currently empty.

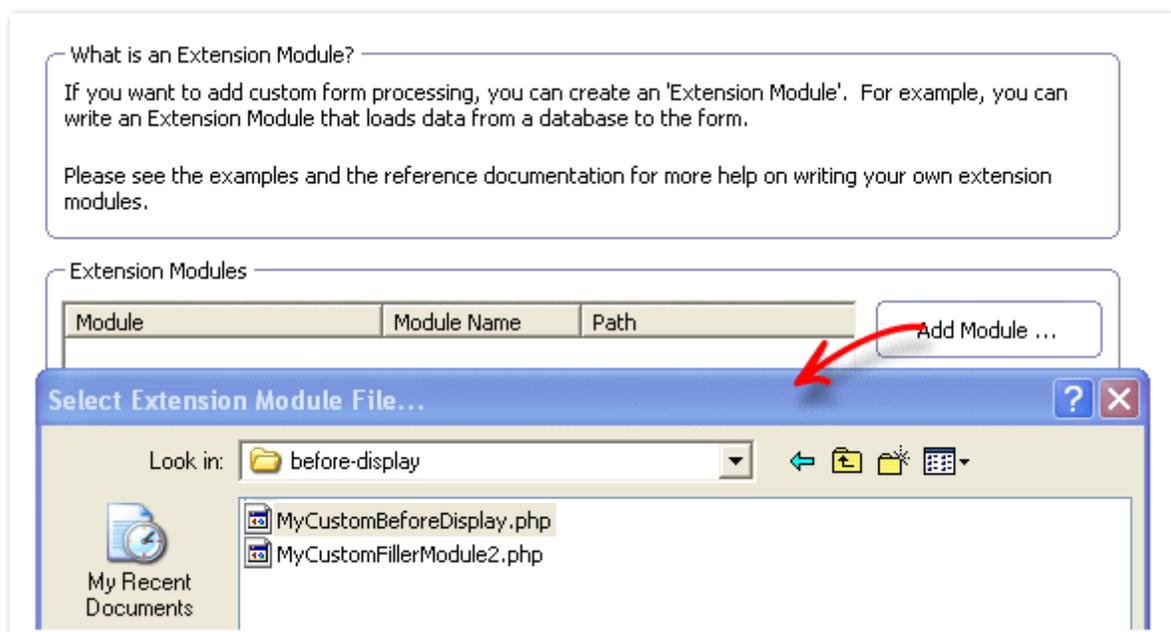
Of course, you can have more complex logic in your extension module.

The Function Reference contains the list of all the functions that you can override in your custom module.

Integrating the module with your form

In Simfatic Forms, first enable extension modules in the Form Processing Options Page.

In the 'Extension Modules' Page you can load your extension modules. Press the 'Add Module' button and select the file that you have created.



You can add any number of extension modules to your form. The overrides will be called one by one (in the order they appear in the 'Extension Modules' page)

Features available to the extension modules

The base class for the extension modules - `FM_ExtensionModule` class contains certain useful member objects.

Logging

You can use the Simfatic Forms standard logger in your extension module. The logged messages will appear in the log if you enable logging in your form. (Enable logging in 'Form Processing Options' page)

To Log an error, call `LogError()` function

Example:

```
$this->logger->LogError("Could not connect to the database")
```

To log an information message, call `LogInfo()` .

Example:

```
$this->logger->LogInfo("Connected to the remote service")
```

Form submission variables

The values submitted in the form are available in the member array `$this->formvars`

Example:

```
$rec = $this->formvars['FirstName'].' '.$this->formvars['LastName'];
```

Handling errors in an Extension Module

To handle a critical error case in your extension module, make a call to the `error_handler` object.

Example:

```
$link = mysql_connect('localhost',  
                    'mysql_user', 'mysql_password');  
  
if(!$link)  
{  
    $this->error_handler->ShowError(  
        "Error connecting to DB:".mysql_error());  
    exit;  
}
```

Sample Extension Modules

There are some sample Extension Modules in the Simfatic Forms Installation folder (usually "C:\Program Files\Simfatic Solutions\SimfaticForms"). Open the sub-folder "scripts\server\ExtensionModuleSamples". You will find some sample extension modules. The same folder contains some form template files as well. These sample modules will serve as a good starting point.

See Also:

[Function Reference](#)

[Update Configuration Settings](#)

['Extension Modules' Page](#)

Function Reference

Override functions

These overrides are called at specific points in form processing.

BeforeStartProcessing()

function BeforeStartProcessing()

This function is called at the very start, before any processing starts.

Override BeforeStartProcessing() function to do any initialization, or configuration setting.



This function is called each and every time the form processor script is invoked.

```
<?PHP
class MyCustomUploadFolder extends FM_ExtensionModule
{
    function BeforeStartProcessing()
    {
        $this->logger->LogInfo("BeforeStartProcessing is
called");

        $this->config->file_upload_folder =
            '../.../..../uploads/form1';

        $this->config->allow_nonsecure_file_attachments = true;

        //you can initialize PHP settings too.
        ini_set("sendmail_from", "me@mywebsite.com");
```

```
    return true;
}
}
?>
```

BeforeFormDisplay()

```
BeforeFormDisplay(&$formvars, $pagenum)
```

This function is called **before** form is displayed.

Override this function to load the form with initial data. To fill the form, set the values in the `$formvars` associative array. For example, if a text field "Name" is to be loaded, set `$formvars['Name']='value'`;

`$pagenum` is the current page being displayed.



The function must return true or false. If the return value is false, the form is not displayed.

LoadDynamicList

```
function LoadDynamicList($listname, &$rows)
```

If you have lists loaded from database, you can override the default functionality and load the list using your own function.

Just override the `LoadDynamicList()` in your extension module.

`$listname` is the name of the list in the form. (you can have any number of lists) .
Return the list in the `$rows` array. as name value pair.

See the example below:

```
<?php
class MyListLoader extends FM_ExtensionModule
{
    function LoadDynamicList($listname,&$rows)
    {
        if($listname == 'dl1')
        {
            $rows[] = array('name'=>'Select
...','value'=>'');
            $rows[] = array('name'=>'Item 1','value'=>'Item
1');
            $rows[] = array('name'=>'Item 2','value'=>'Item
2');
            $rows[] = array('name'=>'Item 3','value'=>'Item
3');
            $rows[] = array('name'=>'Item 4','value'=>'Item
4');
            return true;
        }
        else if($listname == 'dl2')
        {
            $rows[] = array('name'=>'Select
...','value'=>'');
```

```

        $rows[] = array('name'=>'ExItem
1','value'=>'ExItem 1');

        $rows[] = array('name'=>'ExItem
2','value'=>'ExItem 2');

        $rows[] = array('name'=>'ExItem
3','value'=>'ExItem 3');

        $rows[] = array('name'=>'ExItem
4','value'=>'ExItem 4');

        return true;
    }
    return false;
}
}
?>

```

The code above first checks for the list name (d11 and d12 being the list names) and loads the list.

LoadCascadedList

```
function LoadCascadedList($listname,$parent,&$rows)
```

You can custom load cascaded lists by overriding this function. Cascaded lists are lists loaded based on the selection in another list. (See Dropdownlist)

\$listname is the name of the list and \$parent is the item selected in the parent list. You have to return the list of items in the \$rows array. See the example below:

```

<?php
class MyLoadCascaded extends FM_ExtensionModule

```

```
{
function LoadCascadedList($listname,$parent,&$rows)
{
    $this->logger->LogInfo("ext module: LoadCascadedList
$listname parent $parent");
    if($listname == 'dl2')
    {
        if($parent == 'Item 1')
        {
            $rows[] = array('name'=>'Sub Item
1.1','value'=>'Sub Item 1.1');
            $rows[] = array('name'=>'Sub Item
1.2','value'=>'Sub Item 1.2');
            $rows[] = array('name'=>'Sub Item
1.3','value'=>'Sub Item 1.3');
            return true;
        }
        else if($parent == 'Item 2')
        {
            $rows[] = array('name'=>'Sub Item
2.1','value'=>'Sub Item 2.1');
            $rows[] = array('name'=>'Sub Item
2.2','value'=>'Sub Item 2.2');
            $rows[] = array('name'=>'Sub Item
2.3','value'=>'Sub Item 2.3');
            return true;
        }
    }
}
```

```

        return false;
    }
}
?>

```

DoValidate()

```
DoValidate(&$formvars, &$error_hash)
```

Override this function to Validate the final form submission.

The submitted form data is available in the `$formvars` variable.

If there are errors, set the error message in the `$error_hash` and return false.



In a multi-page form, to validate individual pages, override the `DoValidatePage()` function.

The `DoValidate()` function is called only after the last page is submitted.

For example, if the "Email" value submitted in the form is validated error, set the message as shown here:

```

<?PHP

class MyCustomValidationModule
    extends FM_ExtensionModule
{
    function DoValidate(&$formvars, &$error_hash)
    {
        if(!isset($formvars['email']))

```

```
        {
            $error_hash["email"]="Bad Email";
            return false;
        }
        return true;
    }
}

?>
```

DoValidatePage

```
function DoValidatePage (&$formvars, &$error_hash, $page)
```

Override this function to Validate a page, after the page is submitted. ('next' button is pressed).

The submitted form data is available in the \$formvars variable.

If there are errors, set the error message in the \$error_hash and return false.

```
<?PHP
```

```
class MyCustomValidationModule
    extends FM_ExtensionModule
{
    function DoValidatePage (&$formvars, &$error_hash, $page)
    {
        if ($page == 2 && !isset ($formvars['email']))
```

```
        {  
            $error_hash["email"]="Bad Email";  
            return false;  
        }  
        return true;  
    }  
}
```

?>

The code above validates email in the page 3.

 The page index is 0 based. So on first page, \$page value will be 0.

 First check for the page number when you validate using this function.

PreprocessFormSubmission

```
function PreprocessFormSubmission(&$formvars)
```

Override this function to edit the form submission and add your own cleanup or enhancements.

`PreprocessFormSubmission()` is called after validations. You can add or change the submitted form data before it appears in the form submission email or in the database.

Here is an example:

```
<?php
class MyPreproc extends FM_ExtensionModule
{
    function PreprocessFormSubmission(&$formvars)
    {
        $formvars['Name'] = ucwords($formvars['Name']);

        return true;
    }
}
?>
```

The code above updates the Name field so that all names are auto-converted to 'Uppercase first letter'.

(john becomes John).

BeforeConfirmPageDisplay()

```
BeforeConfirmPageDisplay(&$formvars)
```

BeforeConfirmPageDisplay() is Called just before displaying the 'confirm form submission' page (If the option is selected) .

The submitted form data will be available in the \$formvars variable. You can customize the display of the values by modifying the values in the \$formvars array.

For example, the code below makes the 'Name' display in Red if the submission was 'Bob'. Note that the change affects only the display. Not the actual form submission data.

```
<?PHP

class MyConfirmPageCustomModule

    extends FM_ExtensionModule

{

    function BeforeConfirmPageDisplay(&$formvars)

    {

        if($formvars["Name"] == 'Bob')

        {

            $formvars["Name"] =

                "<font color='red'>".$formvars["Name"]."</

font>";

        }

        return true;

    }

}

?>
```

BeforeSendingFormSubmissionEMail()

```
function BeforeSendingFormSubmissionEMail (&$recipient, &$subject, &$body)
```

BeforeSendingFormSubmissionEMail is called just before sending email to the 'recipients' you have configured in the 'form to email' page. If BeforeSendingFormSubmissionEMail() function returns false, the email is not sent. You can override this function to send the email based on a condition. You can modify the body or the subject of the email as well.

```
<?PHP

class MyCustomConditionalEmail

    extends FM_ExtensionModule

{
    function BeforeSendingFormSubmissionEMail(&$recipient,
        &$subject,&$body)
    {
        if(false !== strpos
($recipient,'admin@website.com'))
        {
            return true;
        }
        elseif($this->formvars['SendTo'] == 'sales' &&
            false === strpos($recipient,'sales@mywebsite.com'))
        {
            return false;
        }

        if($this->formvars['SendTo'] == 'support' &&
            false === strpos
($recipient,'support@mywebsite.com'))
        {
            return false;
        }
        return true;
    }
}

?>
```

BeforeSendingAutoResponse()

function BeforeSendingAutoResponse (&\$recipient, &\$subject, &\$body)

BeforeSendingAutoResponse() is called just before sending auto-response (if you have enabled auto-response in the 'form processing options' page).

If this function returns false, the auto-response is not sent. You can change the recipient or the body or the subject of the email.

```
<?PHP
```

```
class MyCustomResponseEmail
    extends FM_ExtensionModule
{
    function BeforeSendingAutoResponse (&$recipient,
                                        &$subject, &$body)
    {
        //The Name part is composed using FirstName
        //and LastName fields
        $recipient = $this->formvars['FirstName'].
            ' '.$this->formvars['LastName'].
            '<'.$this->formvars['Email'].'>';

        $this->logger->LogInfo("New recipient $recipient");

        //update the body of the email with conditional sections
        if(isset($this->formvars['subscribe']))
        {
            $body =
```

```
        str_replace('_REPLACE_CODE_',
                    'Thanks for the subscription too!', $body);
    }
    else
    {
        $body = str_replace('_REPLACE_CODE_', '', $body);
    }
}
}

?>
```

FormSubmitted()

```
function FormSubmitted(&$formvars)
```

FormSubmitted() function is called once the form submission data is finally submitted (After validations and confirmation page, if any).

The form submission data is present in the \$formvars variable. You can handle the final, form submission data in this function.

For example, you can save the form submission to a database table or post the data to a custom web service.

BeforeThankYouPageDisplay()

```
BeforeThankYouPageDisplay(&$formvars)
```

BeforeThankYouPageDisplay() is called just before displaying the 'Thank you page'.

You can customize the display of the values (just like `BeforeConfirmPageDisplay()`) if you are using 'Thank You ' page template and if you have place holder variables. (Like `%Name%`) in the template.

The function is not called if the form is set to redirect to a 'Thank You' page URL rather than displaying a template.

```
<?PHP

class MyConfirmPageCustomModule

    extends FM_ExtensionModule

{

    function BeforeThankYouPageDisplay(&$formvars)
    {
        if ($formvars["Name"] == 'Bob')
        {
            $formvars["Name"] =
                "<strong>".$formvars["Name"]."</strong>";
        }
        return true;
    }
}

?>
```

Update Configuration Settings

You can update the configuration settings by overriding the `BeforeStartProcessing()` function.

Example:

```
class MyCustomUploadFolder
    extends FM_ExtensionModule
{
    function BeforeStartProcessing()
    {
        $this->logger->LogInfo("BeforeStartProcessing is
called");

        $this->config->file_upload_folder =
            '../.../.../uploads/form1';
        return true;
    }
}
```

Here are the configuration settings you can update:

`form_file_folder`

The location to save the files (csv file, the log file etc)

`load_values_from_url`

Whether to allow load values from the URL. For example, if you enter the URL of the form as `myform.php?Name='John'` the value John will be pre-filled in the Name field of the form

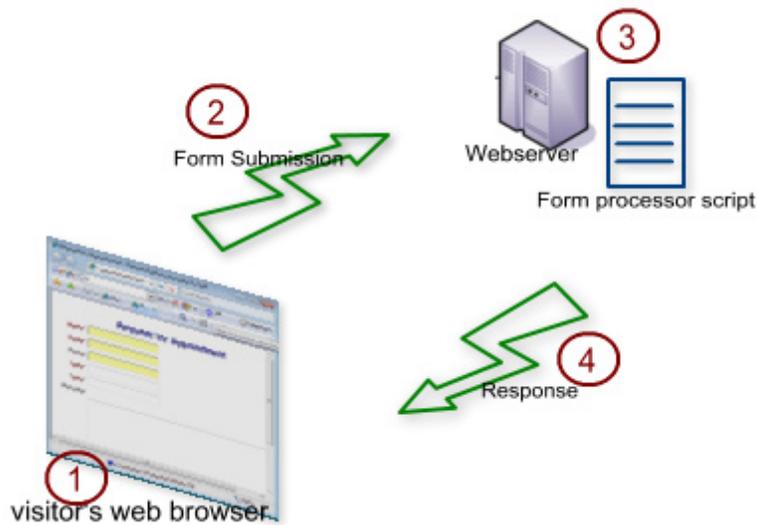
`allow_nonsecure_file_attachments`

In the file upload fields, if the user uploads executable file types, (including PHP script, HTML files and other scripts) it is counted insecure and is not saved by default. Make 'allow_nonsecure_file_attachments' true to allow such file types.

`file_upload_folder`

The custom folder to save file uploads. By default, file uploads are saved under an upload folder under the *form_file_folder* (see above).

4.11 Why do I need to upload form to a webserver ?



A form has two parts: the client side HTML form and a server side form processor script. The client side displays the form and takes the input whereas the server side does the form processing (send email, save form submissions etc). In addition to that, certain components requires support from the server side to display (examples: Captcha, cascaded list).

Going through this tutorial: How does an HTML Form work? will be useful.

Simfatic Forms "Take the code" steps work this way:

- Upload the code.
- Get the URL of the main form script.
- Make the code to embed the form, using the URL of the main script.

"But, I know some form makers that doesn't need any upload !"

Because they host the server side script.

Simfatic Forms generates the code and lets you host the whole form on *your* server. Your forms never expire, you own all the form submission data and you have full control.

"I want to test the form before uploading to my live website !"

You can use the preview feature. Just press the preview button and try your form.

If you want to test your form locally, see how to set up your own local web server

4.12 Error when selecting web publish folder

What is Web publish folder?

Web publish folder is the folder where you upload your files so that those files could be accessed from the web.

For example, let's assume that your web site address is `www.mywebsite.com`. If you have a web page `mypage.html` that needs to be accessed using URL `www.mywebsite.com/mypage.html` you have to upload the file to a folder on the server. The folder where you have to upload the file is called the web publish folder.

Normally the web publish folder is named 'www' 'httpdocs' or something similar. You can get this information from the web-hosting service provider or the administrator.

The Test

When you select a folder and press the 'Test' button (or press 'Next') Simfatic forms conducts a test to see whether the selected folder is the right web publish folder. This is the test:

- Create a temporary html file (let's call it 'abcd.html')
- Upload the file to the selected web publish folder on the server (let's assume that the web address of the server is 'www.mywebsite.com')
- Try accessing the file using URL `http://www.mywebsite.com/abcd.html`

If Simfatic Forms could access the file using the URL, then the test succeeds. Else the test fails.

Troubleshooting

- Press the 'Show Details' button and see what went wrong. The 'Details ' box shows each step that Simfatic Forms did and the result.

- Run the test yourself!

Log on to your website using your favorite FTP client, upload a sample file(say 'abcd.html') to the selected folder and try accessing the file using URL of the form <http://www.mywebsite.com/abcd.html>

4.13 Features that requires database support

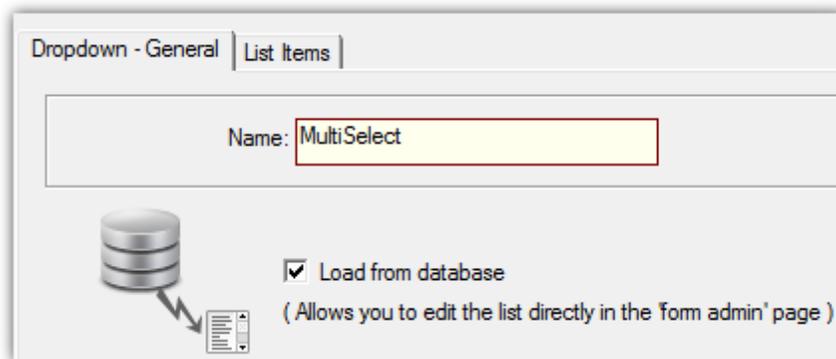
There are Simfatic Forms features (other than 'Save submissions to database') that requires database support. Simfatic Forms will request database login information when any of these features is selected.

The features that requires database support are:

- Save Form submissions to database table
- Paypal support

When Paypal completes the payment processing, Paypal sends a notification (called IPN). The form submission is completed only when the IPN is received. Simfatic Forms keeps the form submission data temporarily in a database table till the IPN is received.

- Dropdownlist or Multi-select loaded from database



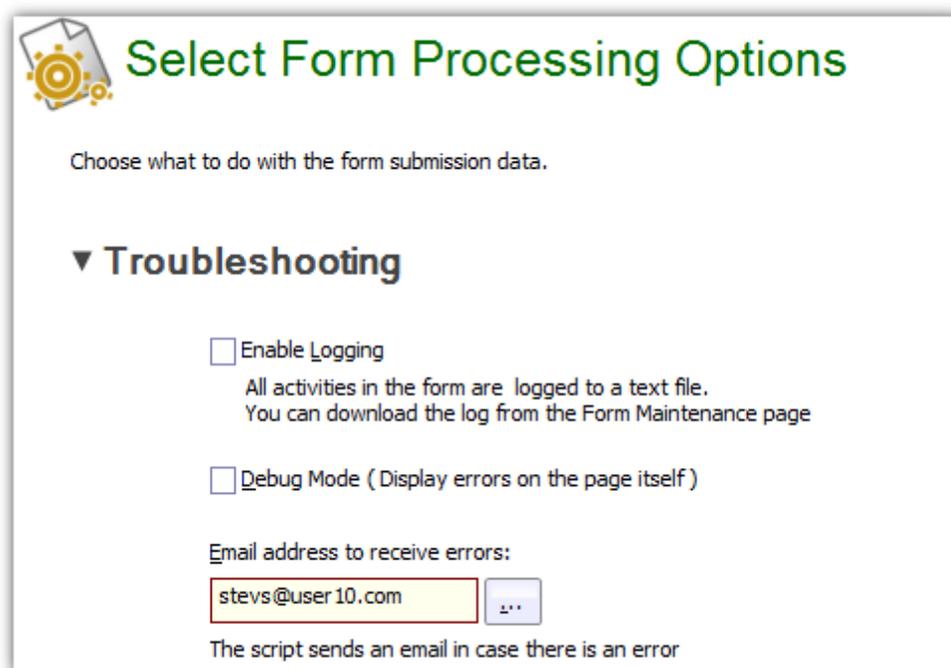
- Cascaded List

4.14 Turn logging off for better performance

Logging and debug mode are useful for troubleshooting in case your form is not working well. Logging shows the path the script is taking. Enabling debug mode shows the errors on the page itself.

Once the errors are corrected and your form is working well, remember to turn those troubleshooting options off. Turning off logging and debug mode helps running your form faster.

To disable logging, go to Simfatic Forms -> 'Form processing options' page -> Troubleshooting section.



 **Select Form Processing Options**

Choose what to do with the form submission data.

▼ **Troubleshooting**

Enable Logging
All activities in the form are logged to a text file.
You can download the log from the Form Maintenance page

Debug Mode (Display errors on the page itself)

Email address to receive errors:

The script sends an email in case there is an error

It is highly recommended to give an email address to receive errors (in case any).

After making the changes, re-upload your form so that the changes takes effect.

4.15 Scared of write permissions to formdata folder

"I am scared of giving 777 permissions to form data folder. Any alternatives?"

If you have selected the features that requires writing to file, then Simfatic Forms requires write permission.

- Give write access only to the PHP/webserver process.

This will require some amount of technical expertise though. You can get help from the server admin as well. For example, if your setup is apache + PHP and if apache is running under the www-data user privilages, give write access to www-data user only.

- Skip the features that requires writing to a file.

These are the Simfatic Forms features that requires writing to a file :

- logging
- save submissions to a csv file
- file uploads

When these features are enabled, Simfatic Forms has to write those files and hence requires write permissions. If you prefer not giving write permissions, just disable these features and it should work fine.

- Relocate the formdata folder.

You can use an alternative location for the formdata folder. Note that the alternative folder should have write permissions.

Steps:

- Create a folder on your webserver (folder that is not accessible from the web)
- Give write permissions to that folder
- Download the formdatafolder extension module
- unzip the extension module and edit the formdatafolder.php in a text editor
change the path to the form data folder
- In Simfatic Forms->Form processing options page, enable extension modules.
Press Next.
- In the 'extension modules' page, press Add module and select the updated
formdatafolder.php file.
- re-upload the form

4.16 Limitations of 'full form code'

There is an option to copy the 'full form code' to embed the form in your page.

You will get the 'raw HTML' code of the form to paste in you page.

Some prefer this mode of embedding the form because they can customize the HTML code further.

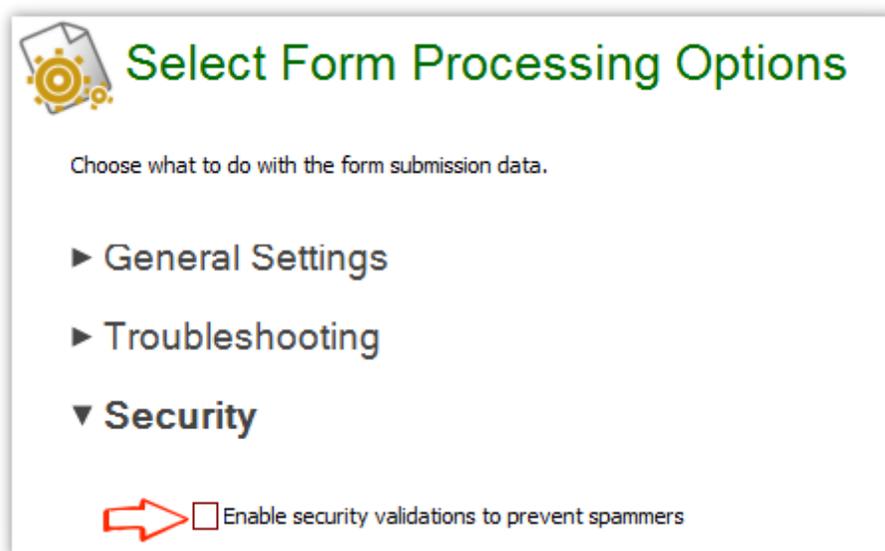
However, there are certain limitations to the raw HTML code embedding.

- There are more chances of your existing page code interfering with the HTML form code.

Open HTML tags/placing the tags in the wrong place can affect the form. Your page may not look broken; but at a later point, the form may not work as expected. Similarly, the JavaScript snippets in your page also can interfere with the form's working.

- The advanced security checks can't work with the embedded full form code.

The security checks depends on validating the form right from the display of the form to make sure that the form is submitted by a normal user rather than an 'auto-bot'. In fact, you have to disable the security checks in the 'form processing options' page to get the full form code.



It is recommended to use the form using JavaScript embed the form code or the iframe embed code

In this case, the form page code is 'sand boxed' from any interference from the parent page. Your form works smoothly.

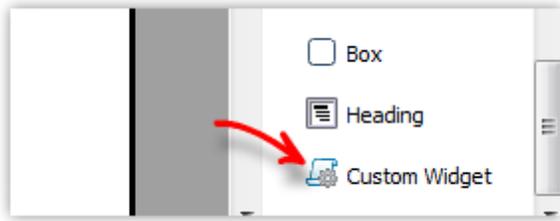
Better customization without 'full form code'

There are ways to customize the generated HTML code other than directly editing the raw HTML form code.

 To initialize a value in your form with a custom value, pass it through the URL .
See: How to pass a value to the form

 To add custom HTML/CSS/Javascript code, use the custom code feature (Draw the form page->Standard tab->custom code button) See: Custom Code Dialog Box

 If you want to insert custom code inside your form, use the custom widget option



Part

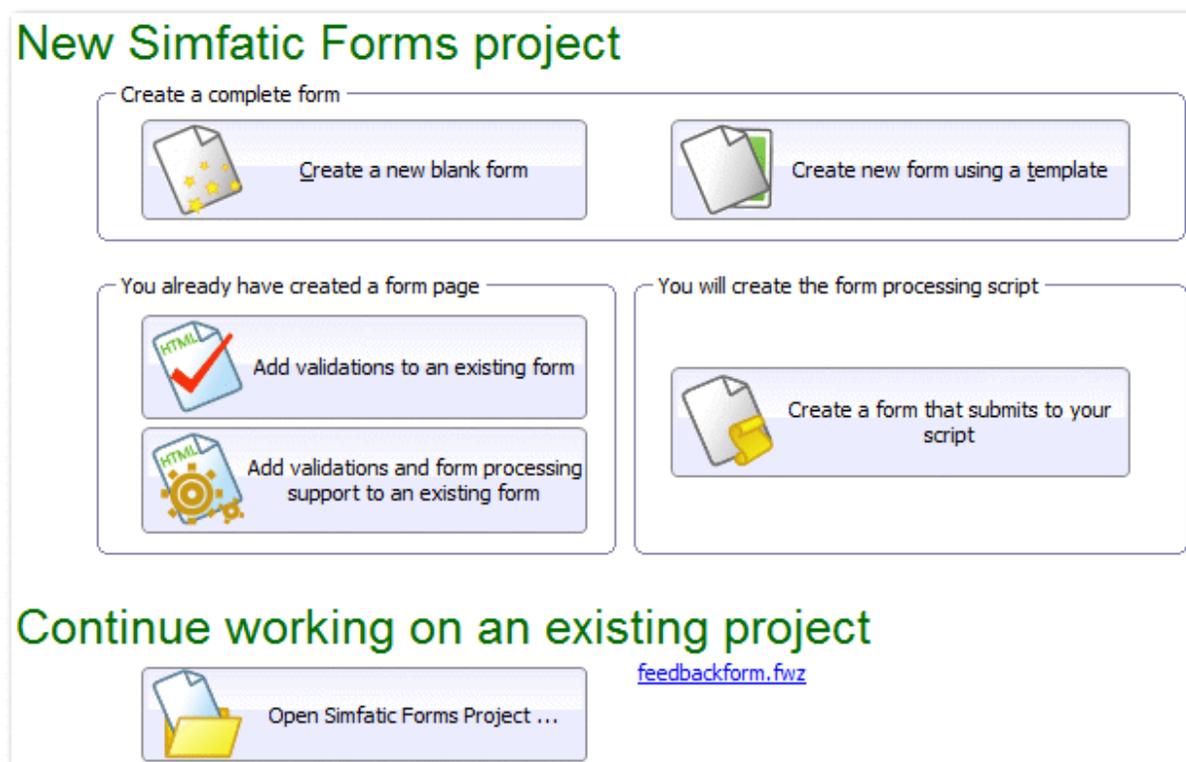


UI Reference

This section documents all the UI screens in Simfatic Forms. In Simfatic Forms press the F1 key to take you directly to the help information for that UI screen.

5.1 The First Page

When you start a new Simfatic Forms project, you see this screen.



1. Create a new blank form

Creates a blank form. You have to add the input elements and then select the form generation options.

2. Create a new form using a template

Frequently required form types are included as form templates. Your new project will be initialized with the input elements from the template. Form templates enable you

to develop your form quickly.

See Create form using a template for more details.

3. Add validations to an existing form

Grabs the form from an existing HTML file. You can create your form in your favorite editor and use Simfatic Forms to read it, and add validations.

4. Add validations and form processing support to an existing form.

Similar to option 3 above. Simfatic Forms reads the form from an HTML page. You can add validations and form processing support to the form and then use Simfatic forms to install the form on your web server.

5. Create a form that submits to your script

In case you want to use your own script to process the form submissions, you can use this option. Use Simfatic Forms' editor to draw a stylish form visually and then use your own script to process the form submissions.

6. Continue working on an existing Simfatic Forms project

While working on a Simfatic Forms form design, you can save your design and options to a Simfatic Forms project file (with extension .fwz) Choose the Save button from the tool bar (or menu item File->Save) to save your form project.

You can continue your work by opening the project file or selecting a recent file from the list.

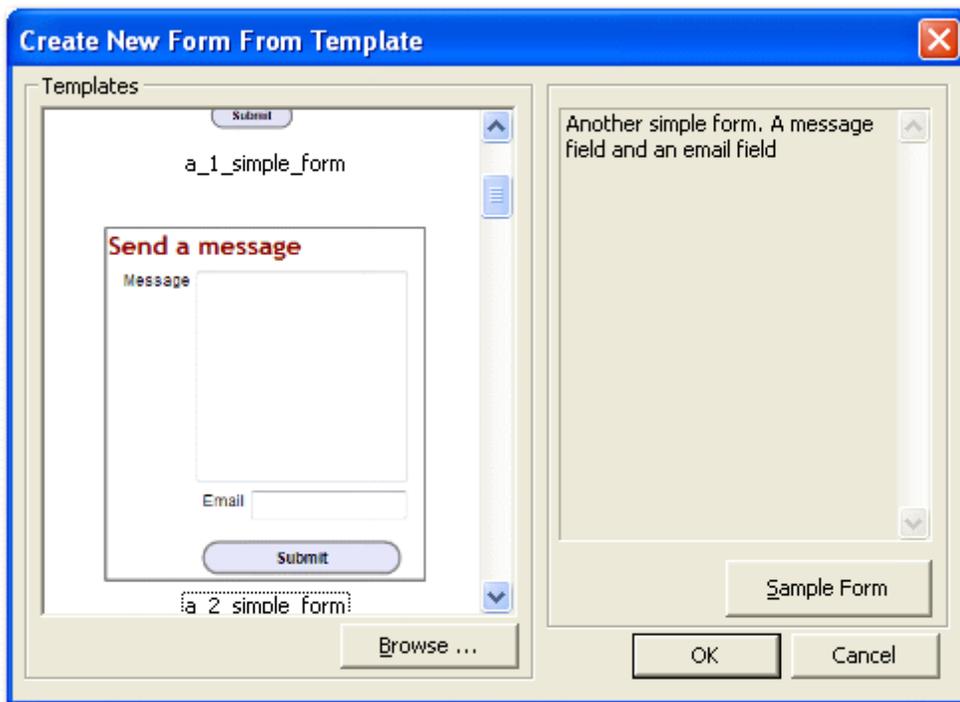
See: Read form from a web page

Create form using a template

Templates provide a quick starting point for creating a form. A form template contains a set of input elements, with associated validations. You can select a template that suits your needs and make necessary modifications to create your form.

In order to create a form from template, click the ' Create a Form using a Template ' option in the First Screen of Simfatic Forms. This opens the ' Select Form Template ' dialog box.

Select Form Template Dialog box



The left side list shows a list of form templates. Select any of them and the right side box shows a description of the form.

You can click on the ' Sample Form ' button to see a sample form created using the template.

Select the template from the list and then press 'OK' to create a form using the template.

5.2 Menu Options

File -> New

Starts a new form project. The currently opened project is closed and a new one is started.

File -> Open

Open an existing Simfatic Forms file. A form project that you create could be saved to a Simfatic Forms file . You can continue working on the same project by opening the saved Simfatic Forms file (file extension is .fwz)

File -> Save

Save the current Simfatic Forms file. You can open this file later and continue with the form project that you were working on.

File -> Save As

Save the current project to a new one. Use this option to create a replica of the current project in a new name.

File -> Save Form as Template

Save the current form as template so that you can create similar forms. You can create forms of generic nature and save it as template. Later for creating forms of similar nature you can use the template.

View -> Toolbar

Toggles between showing and hiding the toolbar.

View -> Status bar

Toggles between showing and hiding the statusbar.

View -> Preview

Shows a preview window containing the form you are developing. Preview allows you to see the form 'live' as you are developing it. The preview menu item is enabled only when you have created at least one input element in the form.

View -> Installation Maintenance

Shows the form maintenance page. Once you have installed a form on the server, you can do maintenance tasks from this page.

This menu item is enabled only after the form is installed on at least one server.

Tools -> Options

Opens the Options & settings dialog where you can update the settings like the default date format.

Help -> Contents

Opens the Help documentation

Help -> Simfatic Forms Online

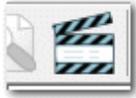
Takes you to the Simfatic Forms web page. Here you can check for any updates to the software.

Help -> About Simfatic Forms

Opens the 'About...' box containing the version and support information

Previewing a form

You can preview the form by clicking on the preview toolbar button



or by selecting menu item View->Preview.

Preview opens the form (in its current state) in your default browser

Options dialog

The options dialog box allows you to update the global level settings of Simfatic Forms. To open the options dialog box, select the Tools -> Options menu item.

General Options

'Thank you' page template file

The HTML file to be used to generate the response page when you press the Auto generate button in the 'Thank You!' Page

Code generation settings

This page contains options and settings related to code generation.

Default code output folder

This is the default folder where the code is generated. For example, if the default output folder is set to 'd:\myformcode' and if you create a new form 'contact', the default code output folder for the form will be 'd:\myformcode\contact'.

Note that you can change the output folder for individual forms in the 'Take the code->Generate Code' step (See
Take the code Wizard)

The DocType of generated web pages

Choose the DOCTYPE of all the web pages generated by Simfatic Forms. The pages

include the templates (like email template and 'Thank You' page) and the form page itself.

For example, if you choose 'XHTML1.0 Strict' as the doctype, all the pages generated will stick to the 'XHTML1.0 Strict' standard.

CSS element positioning style

You can choose whether to have the positioning code be part of the HTML code (inline style) or part of the CSS. Keeping the positioning style in the external CSS would be better since the HTML code will be cleaner and the presentation part is moved to the CSS.

Allow the 'From' field of the email be a value from the form

The 'from' email field of emails should belong to the same domain as the web site where the form is running. This is to ensure that the email passes SPF validations.

For example, if your web site is `www.my-website.com`, the 'From' field of the email should belong to the domain `my-website.com` (example: `myself@my-website.com`)

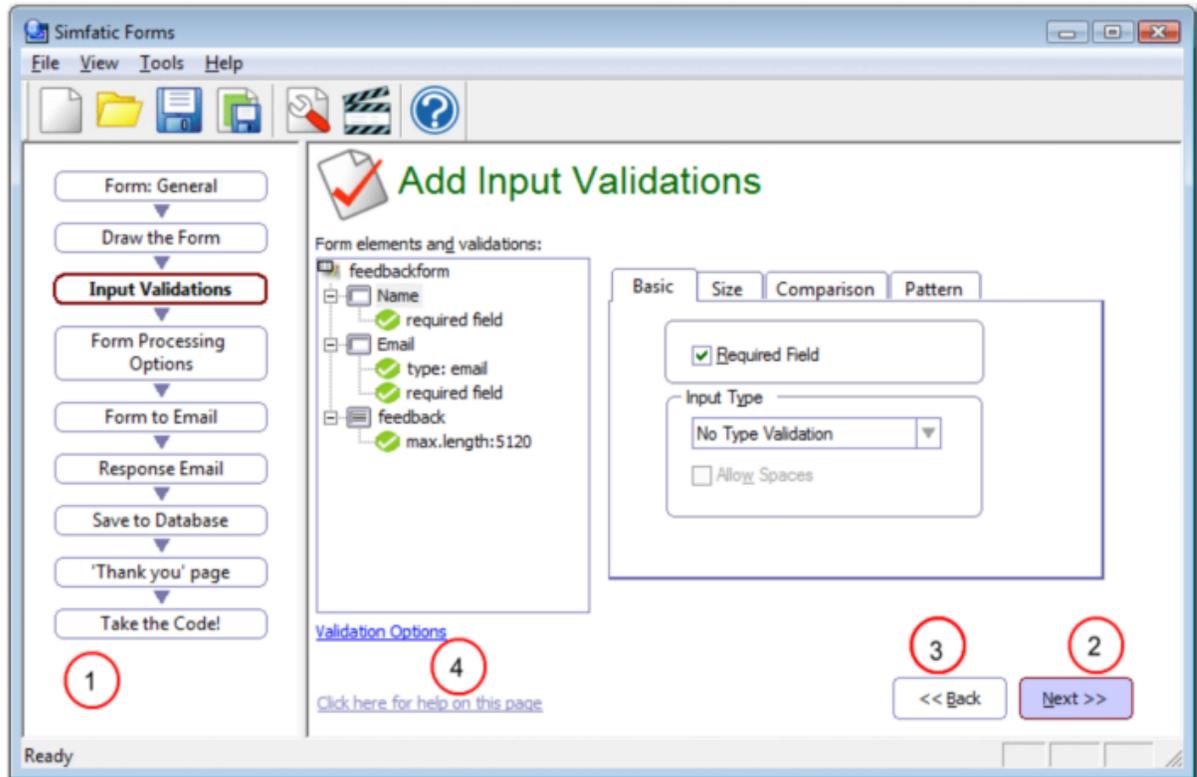
For making it easy to reply to the emails, the 'Reply-To' field is used. (for example, the Reply-To field can be the email submitted in the form; but, not the 'From' field of the email)

Simfatic Forms makes sure that this 'safe SPF' policy is followed, by validating the 'From' field (example: in the Take the code -> Email settings step)

If you want to disable this 'safe SPF' policy, check the option to allow a variable 'From' field. Note however that this can result in the email not getting delivered (because the

email servers that do SPF validation rejects emails)

5.3 The General Layout of the Wizard



1. Steps

The left side shows steps of the Wizard. Once visited, you can go back to any page by clicking the corresponding button in the left-side panel.

2. Next Button

In order to proceed from one step to another, you have to press the 'Next' button. Fill-in the options in each step and press the 'Next' button.

3. Back button

The 'Back' button takes you back to the immediately preceding page.

4. Quick help link

Click on the quick help hyper link on the left bottom side of the page to open help page for the corresponding page.

5.4 General Form Properties Page

In this page, you provide the basic form properties and the general options.

The 'Basic' Tab

Form Name

Name used to identify the form through out the cycle (in the form page, server side processing, in the response mails, form submission data file and so on). It is recommended that you give a unique name for each of your forms.

The form name has to follow certain naming rules

The Look & Feel Tab

You can decide whether you want to customize the ' style ' of the form. (see How to customize the 'style' of the form for more details)

1. Standard

Select this option if you don't want to customize the style and let Simfatic Forms use the standard style for the form.

2. Custom

Select this option if you want to customize the style. If you select this option, the 'Look & Feel' page will allow you to edit the visual style. You can change the font, text color, background color, border color etc. There is an option to load from a set of pre-built styles also.

The 'Form Processing' Tab

Server Side Processing

You can decide how the submitted form submission is processed.

Create only the client-side of the form

A form has two parts: the client-side form that is displayed in your web browser and the server side form processing script. By default, Simfatic Forms will generate both the parts (the HTML form, and the PHP server side script).

If you want to create only the client-side of the form, select this option. This can be useful in two cases:

- You have a server-side script ready (self-coded or custom-coded by third party)
- You want to try just the client-side part. (You can open the generated form page in a browser)

You can uncheck this option at any point later (thus switching back to a complete form).

Form sending Method

The selection determines how the form data is sent to the server. In the "Get" method, the form data is encoded in to the URL. In case of "POST" method the data is sent as part of the message to the server and doesn't appear in the URL.

In most cases, leaving the "POST" method selected will be sufficient.

Localization Tab

There are differences in the way date, number or currency is formatted in different parts of the world. In the localization tab, you can select your locale. The formatting of date number and currency depends on this locale setting.

Localization Settings Dialogbox

You can edit some of the locale settings in this dialog box.

Date Format

You can customize the date format using the formatting tokens below:

Token	Meaning	Example
d	Day of month (no leading zero)	5
dd	Day of month (leading zero)	05
ddd	Day name (abbreviated)	Sat
dddd	Day name (full)	Saturday
M	Month of year (no leading zero)	9
MM	Month of year (leading zero)	09
MMM	Month name (abbreviated)	Sep
MMMM	Month name (full)	September

yy	Year (two digits)	55
yyyy	Year (four digits)	1955

5.5 Look and feel page

You can change the display style of the form in this page.

The layout of the page

The different parts of the form are listed in this page. Click on the item title to expand and change the style of that element.

Page

Change the style of the form page. You can change the back ground colour or add a background image.

Font

The font selected in the page section is inherited by the rest of the form. The same font is used for elements whose font is not customized. This makes it easy to apply same font style across the form.

Please note that some elements (like input elements, error messages, image buttons, progress box) does not inherit from the page font.

Form

This section is to edit the style of the form here. You can align the form to left,right or center in the page. The background style also can be edited.

Labels and Headings

The text style of labels or headings can be edited here. If you select 'inherit' for the font, the Font selected for the Page is applied.

If you leave the background 'none' the label/heading will be 'transparent' and the form/page will be seen through.

The text colour of the label can be edited in the font style

Textboxes

In this section, you can edit the style of single-line textboxes. Choose 'Apply the default style' option if you don't want to customize the style of the textbox.

Multi-line Textboxes

This section is to edit the style of multi-line text boxes (also known as text area).

Dropdown lists

You can customize the font/background color of dropdown lists in this section.

Image button style

The style of the buttons (submit/reset) when the 'image button' option is selected (the default option) for the button.

Validation message style

The style for the error box that is displayed when there are validation errors. This style is used to display the validation errors next to each input element in a movable error message box.

Load a pre-configured style or save as a new style

Instead of editing the individual style elements, you can load a theme: a pre-configured style.

Click the 'Load style' button. A 'file open' dialog opens where you can open a form style file (with file extension: '.fwsf') .

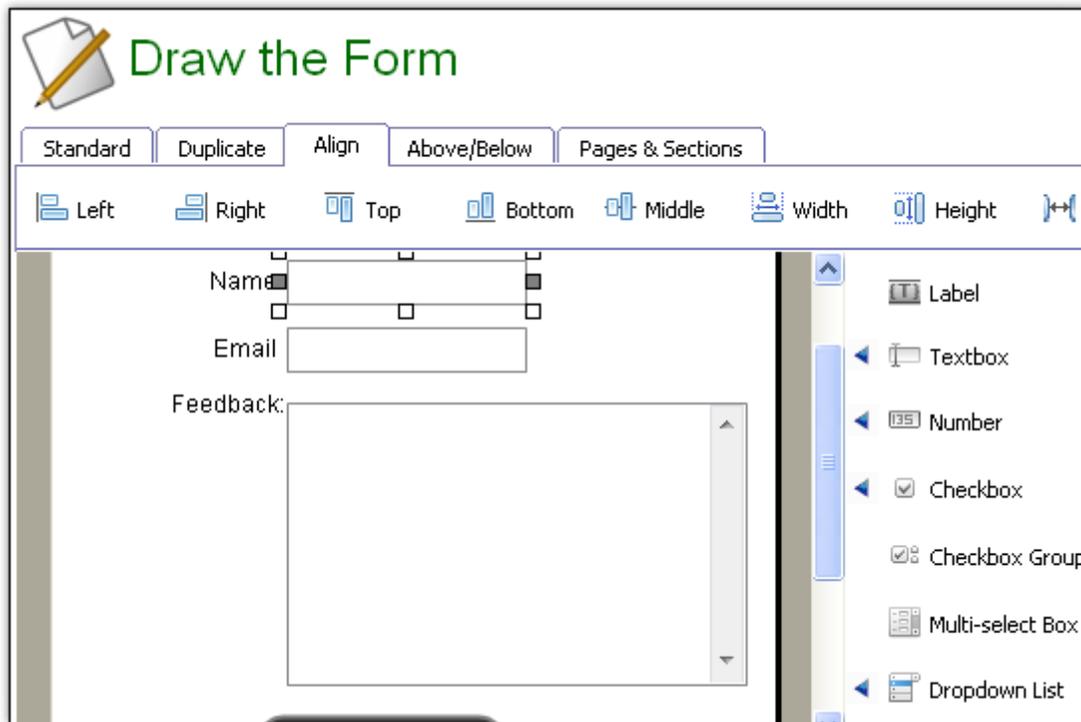
Similarly, you can save your customized style to a file. Click on the 'Save style' button and select the location of the file. Later, you can load your customized style to your form from the file.

See Also:

How to customize the Look & Feel of the form

5.6 'Draw the form' Page

In the 'Draw the form' page you can visually design the form. You can place the form elements and arrange them in the way you want.



See Also:

[How to display elements conditionally](#)

[How to keep your form style consistent with your website style](#)

[Prefer File uploadbox rather than Native file upload](#)

General design operations

Placing a form element on the form

Press the button for the element in the right side to place an element of that type on the canvas. You can move and resize the element using the handles.

Updating the properties of an element

Select the element. Right click on the element and choose 'properties' from the context menu. Alternatively, you can double-click on an element to open the properties dialog box.

Tab order

Pressing the tab order button opens the Tab order dialog box where you can rearrange the tab sequence (The sequence in which the focus shifts from control to control when the 'tab' key is pressed).

Find Element

Press the Find Button ('Standard' tab) to search for any element by name. See Find Element Dialog

Cut, Copy, Paste and Duplicate

The buttons for Cut, Copy and Paste are provided on the top side of the drawing page ('Duplicate' tab).

The duplicate button can be used to quickly make duplicate copies of the selected elements.

Undo, Redo

You can undo the operations multiple levels by pressing the undo button. (Keyboard shortcut Ctrl+Z).The undone operation can be re-acted by pressing the redo button. The undo and redo buttons are in the 'Standard tab)

Grid

Grid enables better placement of the controls. Press the Grid button to enable grid. When the grid is enabled, the controls drawn on the form snaps to the grid while sizing and while moving.

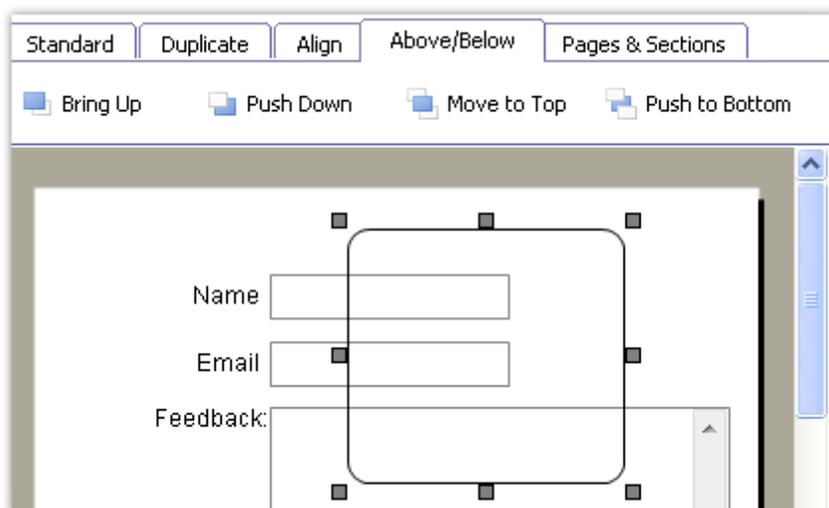
Align

The Align buttons (Align left, right, top, middle or bottom) helps in arranging the elements. Select multiple elements and press one of the align buttons. This places the elements aligned to the selected side.

There are buttons to make the size (width or height) same as well.

The align operations tries to align with the **First selected** element in the group.

Above/Below



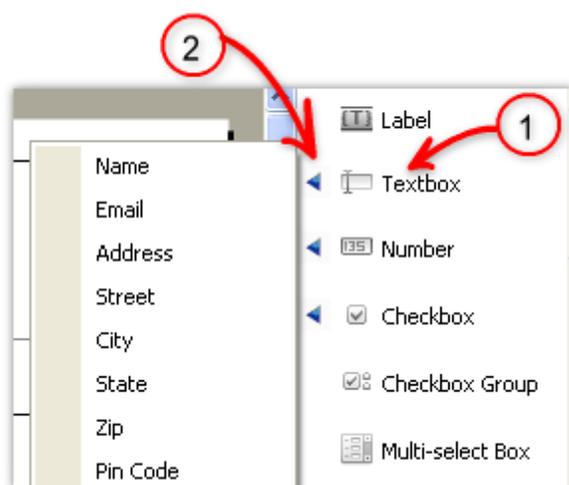
The elements placed on the form can overlap each other. The element above is higher in the z-order (being up in the z axis). When you want to bring an element to the top, press the 'Move to top' button.

For input elements, it is important that no other element overlaps the input element. The input element overlapped by another element can not be selected or edited in the online form. This can cause potential confusion.

When you want to use an image in the background, do 'Push to Bottom' .

Input Elements

This group of buttons lets you add input elements to the form.



Press the tool button (1) to place a new element on the form.

The small-quick-button (2) just to the right of the Input element buttons is to add frequently required input elements (for example, Name, Email or Address)

Pages and Sections

The Pages & Sections tab contains controls to add/delete/modify the form pages.

Form sections can be used to hold conditional parts of the form that can be shown/hidden based on conditions (see How to display elements conditionally). Press the new section button to add a new form section.

See also: Page Settings Dialog Box

Element properties

In the 'Draw the form' page, double-click an element to open the properties dialog box. The pages in this section give the details about property box for each element type.

Label properties

The label element is used to add descriptive text to the form. Press the label button to place a label on the form. Double-click a label element to open the label properties box.

The fields in the dialog box

The text that appears in the label.

Label for

From the drop down, you can choose the element that this label is intended for. This is a standard HTML way to indicate the relation between the label and the input element. You can use the Find element button to search for the element as well.

Horizontal alignment

Determines how the text is placed horizontally within the label box. Left aligned, right aligned or centered.

Vertical alignment

Determines how the text is placed vertically within the label box. Aligned to the top, bottom or middle.

Style

The style tab provides options to edit the style of the label. You can choose to go with the default style (customizable in the Look & Feel page) or edit the style of the current label.

Textbox properties

Double-click the text box element to open the properties dialog box.

The fields in the dialog

Name:

The name is used internally to identify the field.

You need to follow certain Naming Rules

Default Value

The initial value that should appear in the text box.

Auto-hide default value

If this option is selected, the default value is hidden when the user focuses in this input field

(By clicking inside the input field or tabbing to the field)

Password field

Check this if you want to create a password field.

Number Field properties

A number field accepts only decimal numbers (like: 2000.00,13.14,-3456.44, or 1000,000.00) Press the 'decimal field' button to place a decimal field on the form.

Decimal fields are ideal for using in Calculation Fields.

Decimal field properties

Name

Name of the Decimal field. The name is used internally to identify the Decimal field. You need to follow certain Naming Rules

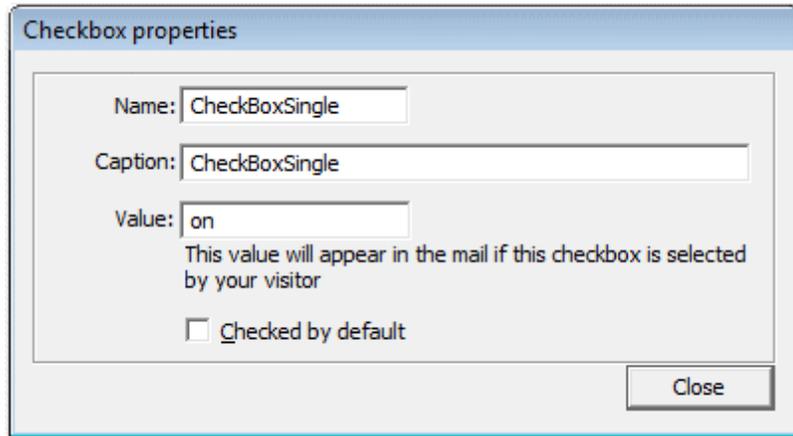
Default value

Provide the number that should appear in the decimal field by default.

Checkbox properties

Check box is ideal for Yes or No questions.

Double-click on a checkbox element to open the properties dialog box.



The fields in the dialog

Name of the check box.

The name is used internally to identify the field.

You need to follow certain

Caption

The text that is displayed next to the check box.

Value

The value that you will see (in the form submission email, File, or database) if this check box is selected by your visitor.

Checked by default

Whether or not to have the check box checked initially.

Checkbox group properties

Create a checkbox group to allow your visitors select multiple options. To create a check box group, press the 'checkbox group' button.

The fields in the dialog

Name of the check box group in which this check box is a member. You can rename the checkbox group by pressing the rename button. To make the check box belong to another group, choose the Checkbox group from the drop down.

The name of the checkbox group need to follow Naming Rules.

Label

Label for the selected check box item. The label appears in the form next to the check box.

Value

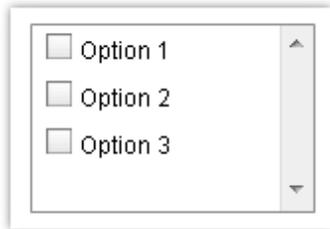
The value that you will see (in the form submission email, File, or database) if this check box is selected by your visitor.

Checked by default

Whether or not to have the check box checked initially.

Multi-select box properties

Multi-select box allows multiple selections from the available choices.



The properties box of Multi-select box is very similar to that of Dropdownlist.

The user can choose one item from a number of available choices in a dropdown list (Also known as pull down menu or combo box)

Name

Give a name to the multi-select list. The name is used internally to identify the multi-select list. You need to follow certain Naming Rules

Load from Database

The list items will be saved in a database table. When the form is displayed, the items are loaded from the database table. This can be handy if the list needs to be updated frequently. You can update the list box directly from the Form Admin Page.

Cascaded List

Cascaded lists are required when one list is loaded based on another list. For example, Country list and state list. When a country is selected in the parent dropdown list, the state list is loaded accordingly.

List Items

In the list items tab, you can enter the items for the list. (One item per line.)

For example:

Select ...

Red

Blue

Green

Magenta

Black

etc.

See some more examples here: [Filling the List items](#)

Load Items from another List

When more than one lists in the same form shares the same list of items, this option becomes handy. For example, if you have two 'country' drop downs in your form, you can make the second list copy the items from the first 'country' list. This will greatly improve the form load time.

Dropdownlist properties

The user can choose one item from a number of available choices in a dropdown list (Also known as pull down menu or combo box)

Name

Give a name for the dropdownlist. The name is used internally to identify the

dropdown list. You need to follow certain Naming Rules

Load from Database

The dropdown list items will be saved in a database table. When the form is displayed, the items are loaded from the database table. This can be handy if the list needs to be updated frequently. You can update the list box directly from the Form Admin Page.

Cascaded List

Cascaded lists are required when one list is loaded based on another list. For example, Country list and state list. When a country is selected in one drop down, the list of states in that country is loaded in another dropdown.

List Items

In the list items tab, you can enter the items for the list. (One item per line.)

For example:

Select ...

Red

Blue

Green

Magenta

Black

etc.

See some more examples here: [Filling the List items](#)

Load Items from another List

When more than one lists in the same form shares the same list of items, this option becomes handy. For example, if you have two 'country' drop downs in your form, you can make the second list copy the items from the first 'country' list. This will greatly improve the form load time.

Filling the List items

This applies to dropdownlist and multi-select listbox. Both of these elements allow free editing of the list items.

Here is the simple format to follow while creating the list

In the 'List Items' tab you can enter the items for the list. The simple format is one item per line, like this:

```
Select ...  
Red  
Blue  
Green  
Magenta  
Black
```

If you want the values to be different from the item name, separate it with a comma

Example:

```
Select ...,0  
Red, #f00  
Blue, #00f
```

```
Green, #0f0
```

```
Magenta, #f0f
```

```
Black, #000
```

etc.

If the item name or the value needs to contain comma, place the item in double-quotes:

```
Select ...,0
```

```
"Red, Green, Blue", 1
```

```
"Purple, Yellow, White", 2
```

Having items selected by default

If you want some items selected by default, have a third item as selected or y

Example:

```
Magenta, #f0f,selected
```

```
Black, #000
```

or, to keep the name and value the same,

```
Green
```

```
Magenta,,selected
```

```
Black
```

Or

Green

Magenta,,y

Black,,y

White



For dropdownlist, only one item can be selected at a time. If you have selected more than one items, the first item is taken as selected.

Using Certain Letters (, and ") in the List

If the item itself contains comma, enclose the item in double quotes (").

Examples:

```
"$2,000"
```

```
"10,000",10000,selected
```

```
"12,999","12999",selected
```

```
"square, rectangle, random shape"
```

If the item contains double quotes("), use two quotes instead of one and enclose the item within double quotes.

For example, if you want to add an item 12"3' to the item list, it should be "12""3' "

More Examples:

```
"91""2' "
```

```
"9,1""2' "
```

```
"9.99"" "
```

```
"The ""Real"" Experience"
```

Cascaded Lists

Cascaded lists change the items when its parent list's selection is changed.

Suppose the parent list contains items like this:

Select ...

Parent 1

Parent 2

Parent 3

Parent 4

The cascaded list can contain items like this:

Select ...

Parent 1>

{

Item 1.1

Item 1.2

Item 1.3

Item 1.4

}

Parent 3>

{

Item 3.1

Item 3.2

Item 3.3

Item 3.4

}

Fixed 1

Fixed 2

Fixed 3

Notice that not all the parent items have child items in the cascaded list. You can have items that will always be present, irrespective of the selection in the parent list too. (Such as the Fixed 1, Fixed 2 and Fixed 3 above)

Using comma and double-quotes (, and ") in the List

if the parent item has , or " (comma or double-quotes), you have to enclose the whole item in double quotes.

To escape double quotes in the string, add another double-quotes: See the example below:

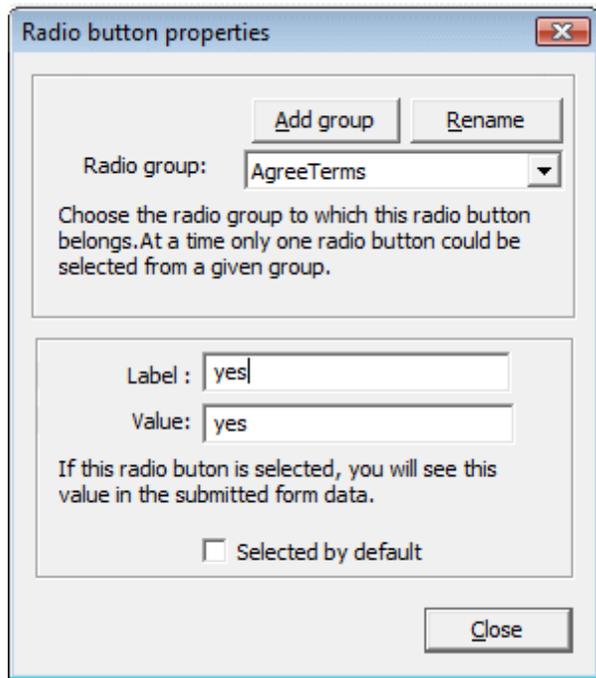
```
" 5"" 6' >"
{
none
" 1"" 3' "
" 2"" 7' "
}
```

Radio button properties

If you want your visitor to select one option from many available choices, use a radio group.

Press the button labeled 'Radio button' to add a radio group to the form.

Double click on the radio button to open the properties dialog box.



Grouping the radio buttons

The individual radio buttons does not make sense until the radio buttons are grouped.

In order to make the radio buttons belong to a group, open the properties of the radio button (by double-clicking) and then choose the 'Radio group'. You can rename the group by pressing the 'Rename' button.

The fields in the dialog

Name of the radio group.

The name is used internally to identify the radio group. You need to follow certain Naming Rules

Press the rename button to rename the radio group.

Press the 'Add group' button to add a new radio group.

Label

Label is the text that appears next to the radio button.

Value

Value is the identification of the selection. The value appears in the form submission data against the radio group name.

For example, suppose the radio group is asking the agreement to the terms and conditions with name 'do_you_agree' and the values are 'yes' and 'no'. If your visitor selects 'yes' then the value will appear as do_you_agree=yes

Selected by default

Choose this option if you want the selected radio item to be selected initially by default.

Multi-line properties

A multi-line text box allows you to gather a larger block of text from your visitor. (Could be an address, an article or a feedback.)

The fields in the dialog

Name

Name of the multi-line textbox. The name is used internally to identify the multi-line.

The name should follow certain Naming Rules

Default text

The text that should appear in the multi-line text box by default.

Auto-hide default text

If this option is enabled, the default text is hidden when the user sets focus in the field. The default text is shown again if the user switches focus without filling the field.

File Uploadbox Properties

The Enhanced File Upload Box gives a convenient file upload interface to attach files to the form submission. The user can upload more than one files and it shows the progress of the file upload. Here are the properties of the Enhanced File Upload Box widget:

Name

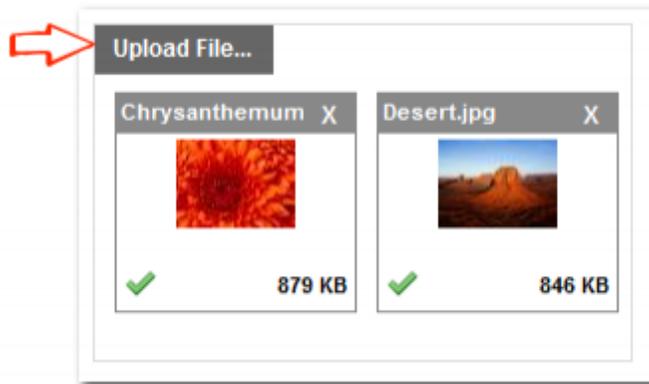
Provide a unique name for the file upload field. The name should follow certain Naming Rules

Labels

You can edit the labels for the different parts of the upload widget here.

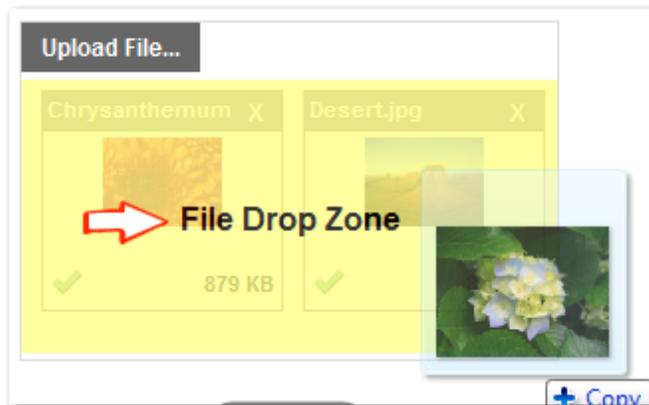
Button Label

The label for the button on the upload widget.



Dropzone Label

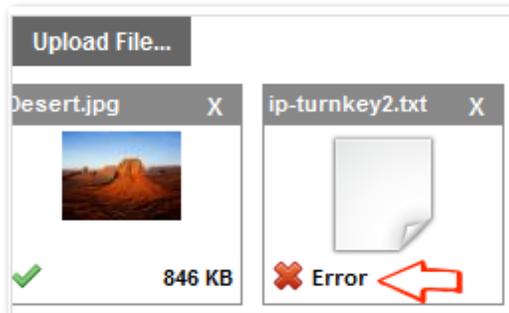
When a file is dragged on to the form page, an area is displayed where the user can drop the file. You can change the label of the dropzone here.



Error box label

The file uploads are validated as each file is added to the upload widget. If there is an error, an 'Error' text is displayed. You can change the 'Error' label here. Note: the error message for the individual validations can be edited in the input validations

page.



Native file upload properties

The file upload box allows your visitors to attach file along with their form submission.

Press the 'native file upload' button to add a file upload box to the form.

The fields in the dialog

Name

Name of the file upload box. The name is used internally to identify the input.

You need to follow certain Naming Rules

See Also: Prefer File uploadbox rather than Native file upload

Hidden field properties

Hidden fields are required if you want to pass a value to the server, that is not entered by the visitor. Usually the hidden field contains a setting or a configuration value.

Press the 'Hidden Field' button to add a hidden field to the form. Hidden field is represented using a dotted rectangle.

Name

Name of the hidden field. The name is used internally to identify the field. You need to follow certain Naming Rules

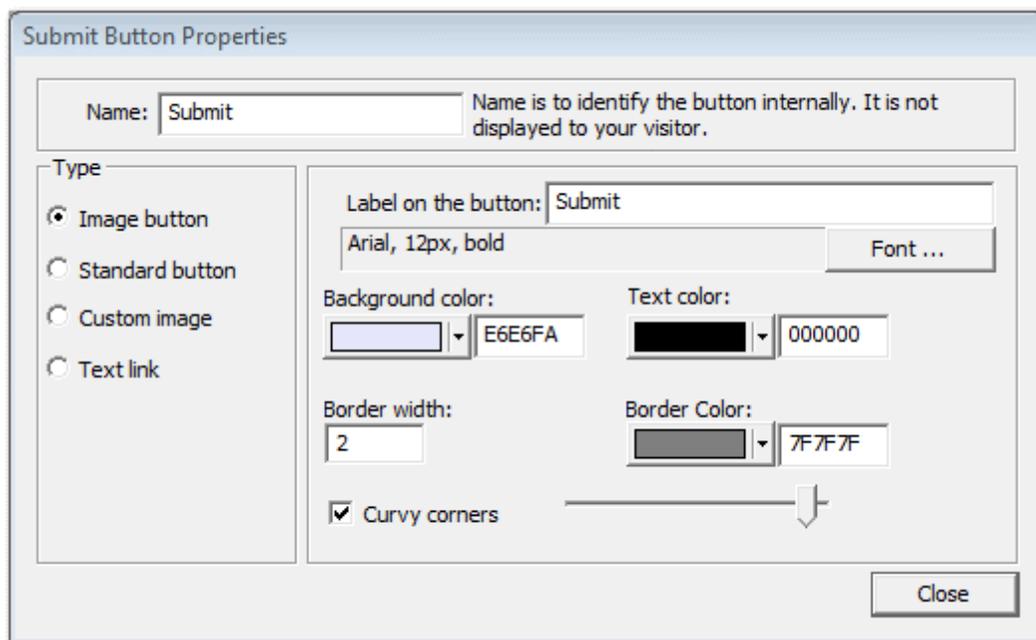
Value

The value of the hidden field.

See Also: [How to pass a value to the form](#)

Submit button properties

Submit button is required in all forms to complete the form submission. You can create a submit button as an image button, as standard button, a custom image or as a link.



The fields in the properties dialog box

Name

Name of the submit button. The name is used internally to identify the button.

You need to follow certain Naming Rules

Button Type

Select the type of the button you want to have. You can select between Image button, Standard button , custom image or a submit hyper link.

Image button

Label on the button

The label text that should appear on the submit button

Font

To change the font of the label text on the button, press the font button and choose the font properties.

Background color

Change the background color of the button. You can press the button to get the drop down box for selecting the color or you can provide the RGB color code in the box provided.

Text color

Choose the color of the label text.

Border width

The width of the border around the button. You can provide 0 to remove the border.

Border color

Choose the color of the border around the button.

Curvy corners

You can have curved corners for the button. Move the slider to control the amount of the curve.

Standard button

This option creates the standard submit button. You have to provide the label that

you want to appear on the button

Image button

Use a custom image as the submit button. Click the browse button and select the image that you want to use as the submit button.

Hyper link

You want to submit the form when a link is clicked. Provide the label of the link in the right side box.

Reset button properties

Pressing the reset button in a form sets all the fields to empty, so that your visitor can start all over again.



Reset is unnecessary for most forms. The danger of the Reset button is that the user might lose all the entered data if Reset is pressed by mistake.

You may consider avoiding the Reset button, if possible.

The fields in the properties dialog box

Name

Name of the reset button. The name is used internally to identify the button.

Button Type

Select the type of the button you want to have. You can select between Image button, Standard button , custom image or a reset hyper link.

Image button

Label on the button

The label text that should appear on the reset button

Font

To change the font of the label text on the button, press the font button and choose the font properties.

Background color

Change the background color of the button. You can press the button to get the drop down box for selecting the color or you can provide the RGB color code in the box provided.

Text color

Choose the color of the label text.

Border width

The width of the border around the button. You can provide 0 to remove the border.

Border color

Choose the color of the border around the button.

Curvy corners

You can have curved corners for the button. Move the slider to control the amount of the curve.

Standard button

This option creates the standard reset button. You have to provide the label that you want to appear on the button.

Image button

This option creates the standard reset button. You have to provide the label that you want to appear on the button.

Hyper link

You want to reset the form when a link is clicked. Provide the label of the link in the right side box.

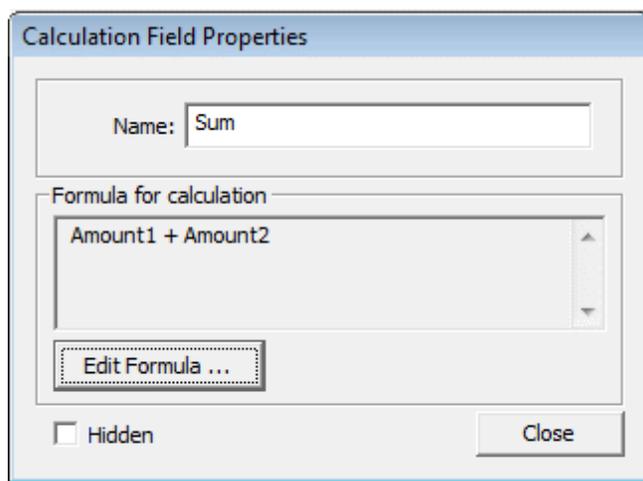
Calculation field properties

A calculation field allows you to have computations based on a formula. The fields in the form can be used as input to the formula.



See a detailed example here: [How to create an online calculator](#)

Press the 'Calculation field' button to place a calculation field on the form.



The fields in the dialog

Name

Name of the calculation field. The name is used internally to identify the field. The name need to follow certain Naming Rules

Edit formula

Press the 'Edit Formula' button to open the Formula editor where you can compose the formula for the calculation.

Hidden

Normally the calculation field is a text box where the result of the calculation is displayed. You can have the calculation result saved in a hidden field. This will especially be handy when you want to calculate a complex formula. A hidden calculation field can be used to keep intermediate results.

Format as currency

The result of the calculation will be formatted as a currency value. For example, 1200.3 becomes 1200.30 Depending on the locale selected in the Form:General->localization tab 1000 separator will be added (1,200.30) and currency symbol will be appended too(\$1,200.30).

See Also:

Calculation Field

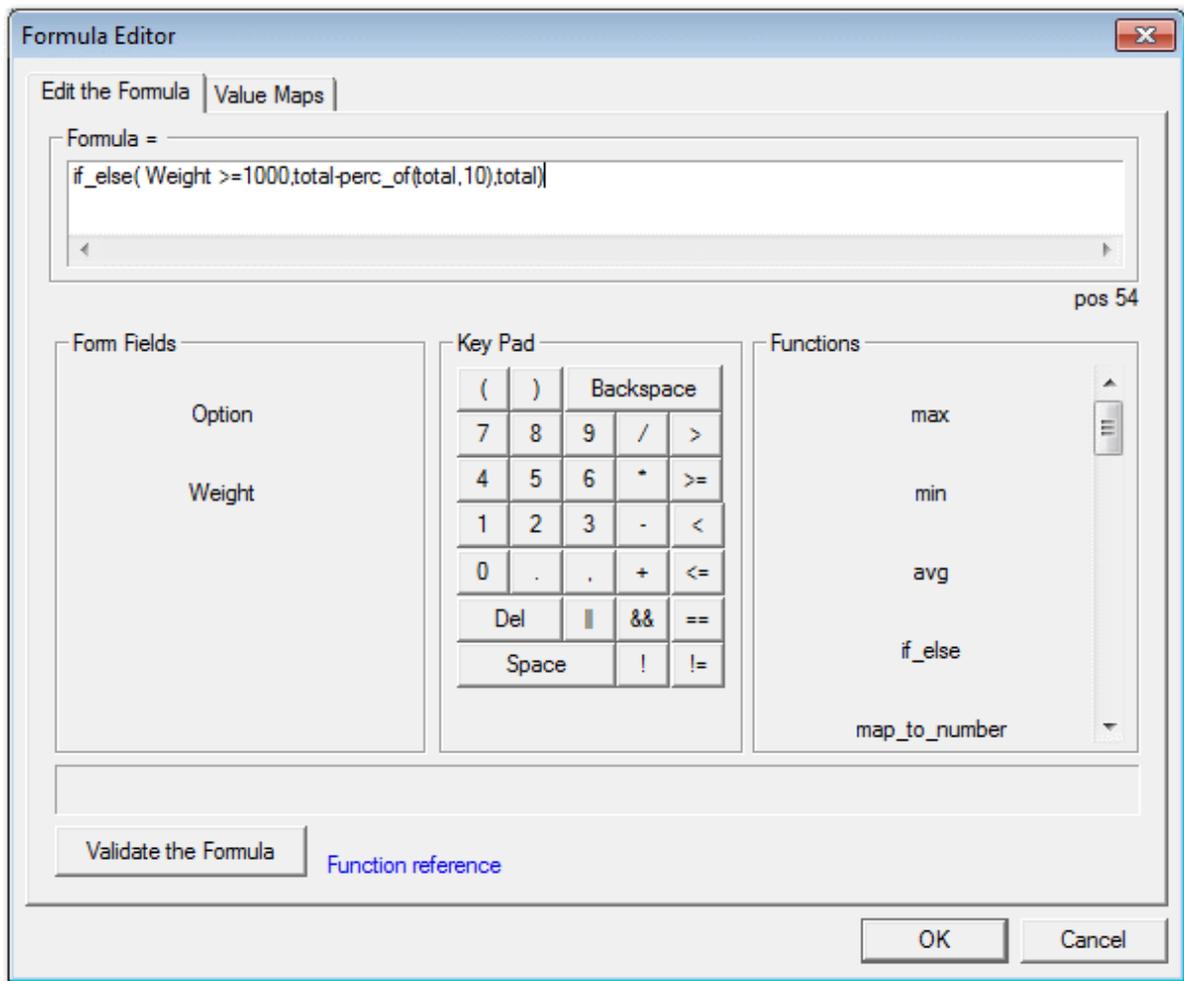
Calculation field : function reference

How to create an online calculator

Formula editor

Formula editor is for composing the formula for the calculation field. Press the 'Edit Formula' button in the calculation field properties dialog box to open the formula editor.

You can use the fields in the form for the calculation. In addition to normal arithmetic operations, you can use a set of in-built functions also in the formula.



Form fields

The 'form fields' box lists the fields from the form. Press a field name to add it to the formula.

Keypad

The keypad contains buttons for the keys that are required while editing the formula. This includes arithmetical operators, comparison operators and digits.

Functions

The functions box contains commonly required mathematical functions and other utility functions.

Press the function name to add it to the formula. Move the mouse cursor above the button to see a short description of the function.

Validate the formula

Press the 'Validate formula' button to validate the formula and display errors if any.

Formula validation will always be done

When you press OK. If there is error, the error is displayed in the box above the Validate the formula button.

Value Maps

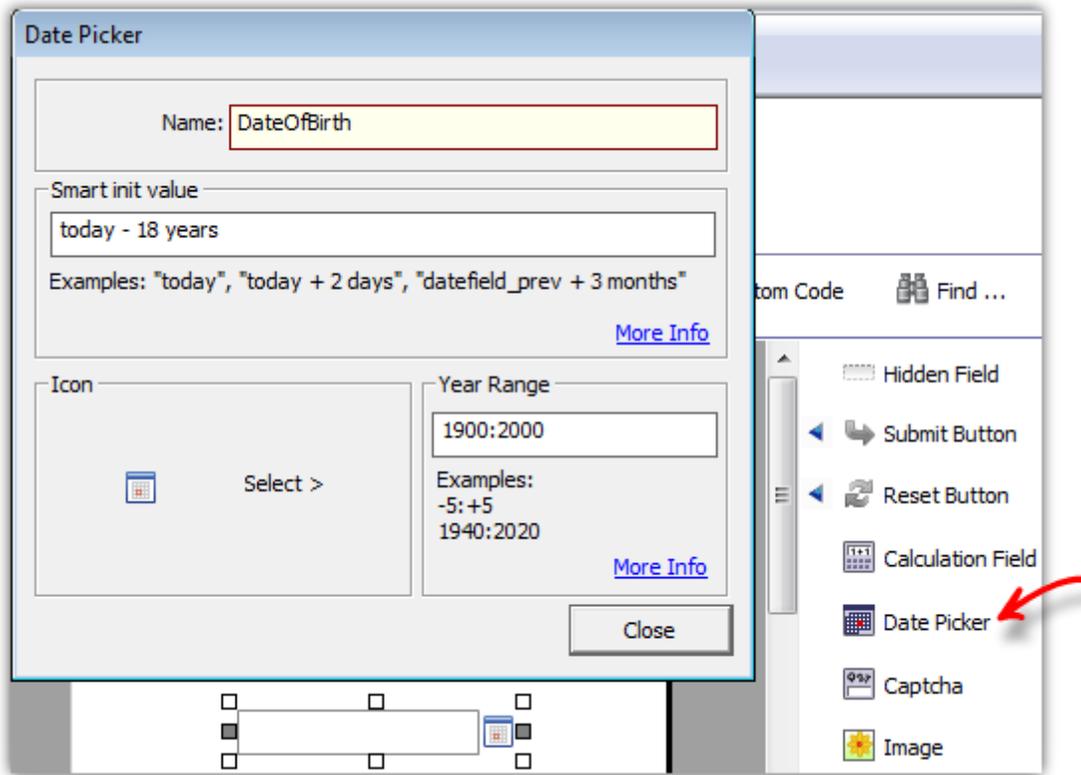
When there are selectable elements (like lists and radio buttons) in the formula, The selected value needs to be mapped to numeric values so that it can be used in the formula. Value maps are to do the mapping. Provide the numeric value for each option.

See Also:

How to create an online calculator

Calculation field properties

Date Picker properties



Date picker lets your users pick a date easily. Moreover, the date picker can be used in calculations also (see Calculation Field).



The date format (MM/DD/YYYY or DD/MM/YY depends on the locale options selected in the Form:General -> Localization tab. The localization tab has option to change the date format as well.

Name

Name of the datepicker field. The name is used internally to identify the field. The name need to follow Naming Rules

Smart Init value

The initial value in the date field. You can have a fixed date (like 27/03/2012) or a dynamic initial value (like 'today' or 'anotherdate + 2 days').

This is the generic format:

```
[today/another_date_field] [+/-] N [days/months/weeks/years]
```

Here are some more examples:

```
today + 1 month
another_date_field - 2 weeks
another_date_field + xx years
```

Where **xx** is a numeric/calculation field

 You can combine a calculation field with the date picker smart init. example:
`another_date + num days` where num is the name of the calculation field.

 If the smart init is based on another date field, (example `another_date + 3 days`) The smart init value is updated every time the user changes 'another_date'. However, Once the user has selected a date, the smart init will not change the value.

 In a multi-page form, you can refer to date field in another page in smart init.

Simfatic Forms will automatically fetch the correct value!

Icon

The icon shown next to the date field box. You can use a custom icon or select from the gallery.

Year range

The year range to be displayed in the datepicker dropdown.

- The year range can be relative to the current date like this: `-5 : +5` meaning years spanning from 5 years in the past to 5 years in the future.
- The year range can be fixed like this as well: `1900 : 2040`

The default is `-10 : +10`

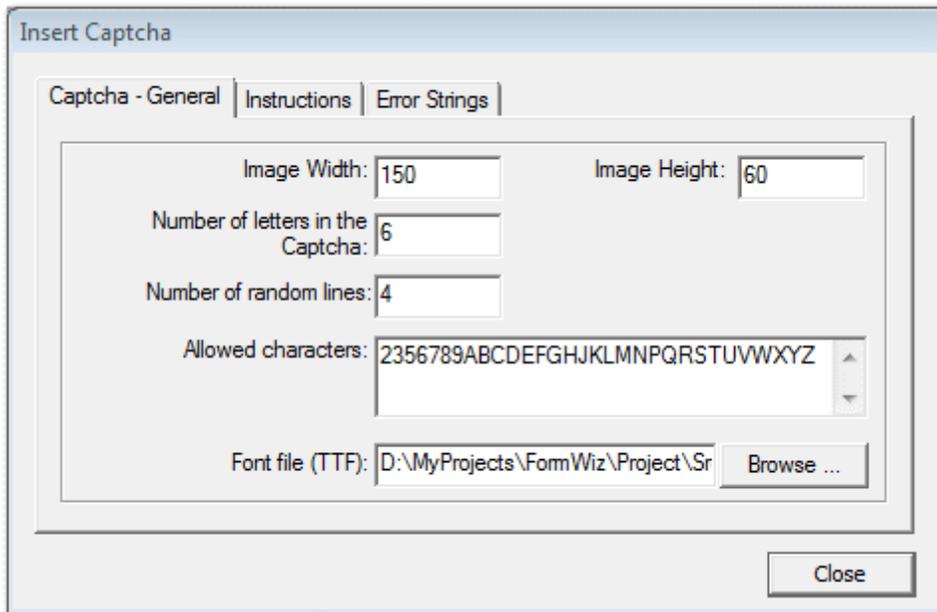
Captcha properties

Captcha is used to prevent spam bots from automatically submitting the form. Captcha ensures that the one who submits the form is human. This is done by displaying a blurred image, that the user has to read and enter in a textbox.



Captcha adds significant difficulty to your form. A good percentage of users will abandon the form owing to the Captcha. Simfatic Forms has other spam control measures.

You may consider avoiding Captcha in your form. See: [Is Captcha required in your form?](#)



The fields in the dialog

General tab

Image Width and Height

The size of the Captcha image.

Number of characters in the Captcha

The number of characters used to create a captcha image. If there are too many characters, it will be more difficult for your visitor to enter the captcha characters.

Number of random lines

While creating the Captcha, a number of random lines are drawn to make it difficult for any automated bots to read the Captcha code. The more the number, the more difficult it will be to read the code. But, it will be difficult even for humans to read the code if there are too many lines. Provide a moderate number of lines for best results.

Allowed characters

The characters used to create the Captcha image. You can avoid similar characters so that to avoid confusion for your visitor. (examples: O an 0, 8 and B etc)

Font file

You can provide a TTF font file to be used to create the characters in the Captcha image. A default font file is provided with Simfatic Forms.

Instructions Tab

You can modify the instructions displayed in the captcha. The default instruction text is provided by default. You can customize the instructions per your needs.

Note that the `{link}here{/link}` in the refresh instruction is the placeholder for the link to refresh the captcha image.

Error strings tab

You can customize the error strings in the error strings tab. The error is displayed on validating the captcha (captcha code was not entered or the entered code does not match)

Image properties

You can place an image in the form by pressing the image button. The Image button opens a menu to choose the image from gallery or from the local disk. Select the image to place it on the form.

Resize the image as per your requirements. Resizing will keep the aspect ratio.

The fields in the properties dialog

Image file

Press the button '...' and choose the image file. The supported formats are jpeg, bmp, gif, and png.

Description

The description of the image that will be provided in the 'Alt' attribute of the image.

Hyperlink Properties

You can put a link to another page on the form using the Hyperlink element.

URL

The complete URL of the page you are linking to.

Link Text

The label text that should appear on the hyperlink.

Link Type

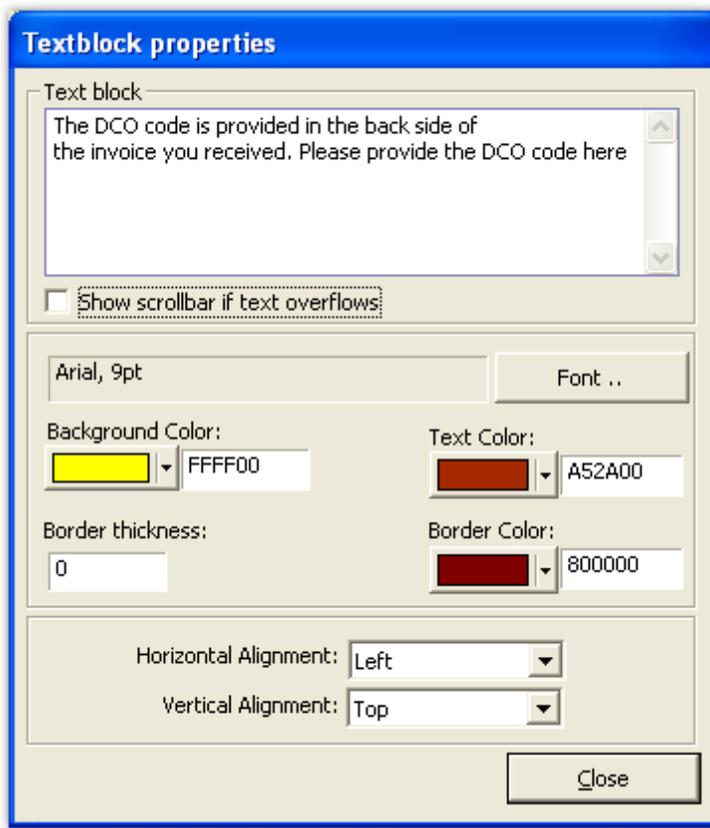
The link can be simple text link or an image. If you choose image for the link, browse and select the image on the right side

Target Window

You can open the link on the same page or in another tab or window. Opening the page in the same window may cause loosing the form data. It is better to open the link in a new tab or in a popup window.

Text block properties

The Text block is used to add descriptive text to the form. Double-click a Text block element to open the text block properties box.



Text block properties

Text block

The text that appears in the text block.

Show scrollbar if text overflows

If the text overflows the bounding rectangle, whether to show a scrollbar or not. You can resize the textblock rectangle. If the text is too big to fit in that rectangle, it will be convenient to have a scroll bar for the rectangle

This option can be used to display long text like Terms & Conditions

See: How to make a Terms & Conditions box

Font

Choose the font properties (font face, size etc) for the text block by pressing the 'Font' button

Background Color

You can select the background color of the text block in this color selection button. To change the color, press the down arrow in the color selection button. If you choose 'None', no background color is applied and the text block box will be transparent.

Border thickness

Determines how thick the border around the text block is. Give 0 if you don't want border.

Border color

Choose the color of the border.

Text color

The color of the text in the text block.

Horizontal alignment

Determines how the text is placed horizontally within the text block box. Left aligned, right aligned or centered.

Vertical alignment

Determines how the text is placed vertically within the text block box. Aligned to the top, bottom or middle.

Box Properties

Box can be used to show visual grouping of the form elements.

Doubecllick on the box element to change the properties

Title

The heading text to be shown on top of the box. For example:

A screenshot of a form box titled "Contact Details". The box has a rounded rectangular border and contains two input fields. The first field is labeled "Name" and the second is labeled "Email".

Style

Here you can customize the style of the box.

Border

You can select the width and color of the border of the box

Background

The background can be transparent, or can be a solid colored background. You can select a background image as well

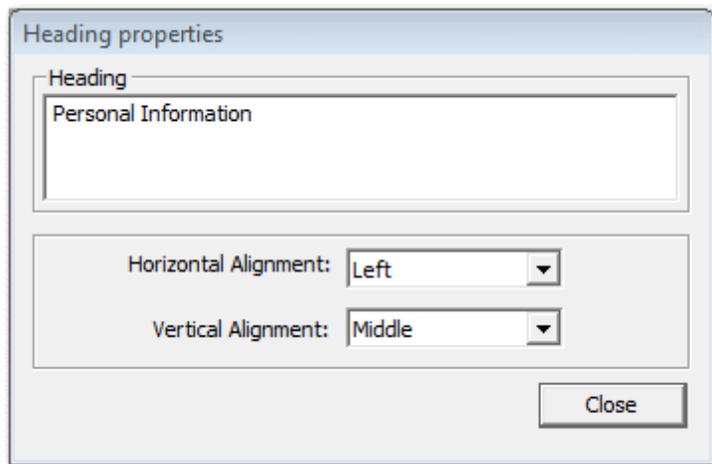
Title Background Color

The title can be transparent or can be a solid color background

Heading properties

Heading properties

You can use headings to group elements in the form and to create sub sections. Double-click a Heading element to open the Heading properties box.



The fields in the dialog box

The text that appears in the Heading.

Horizontal alignment

Determines how the text is placed horizontally within the Heading box. Left aligned, right aligned or centered.

Vertical alignment

Determines how the text is placed vertically within the Heading box. Aligned to the top, bottom or middle.

Custom Widget Properties

Custom widget allows you to create put your own widgets or third party widgets downloaded from the internet on your form.

You can use a custom widget to put your own HTML code as well.

 **Note:** You have to make sure that the custom code you put in the Custom widget is valid. Invalid code might cause the form stop working.

Custom HTML Code

Put the custom HTML code here.

Name

The name of any input elements in the custom HTML code. You can use this name in the other steps (like form to email) to fetch the value from the field.

For example, if the custom HTML code is this:

```
<input name='widgetinput' type='text' />
```

You can use the input field with name *widgetinput* in the later steps.

If there is no input element, you can skip this field.

Javascript file for the custom widget

If your custom widget has Javascript handler, select the javascript file in this step. If there are more than one Javascript files, you may have to combine them in to one. Note that you can include more Javascript files through the Custom Code Dialog Box option as well.

Stylesheet (CSS) for the custom widget

If your custom widget has custom style, you can provide the path to the stylesheet path here. If the CSS code uses custom images, put it in an images sub folder. Simfatic Forms will copy those images as well.

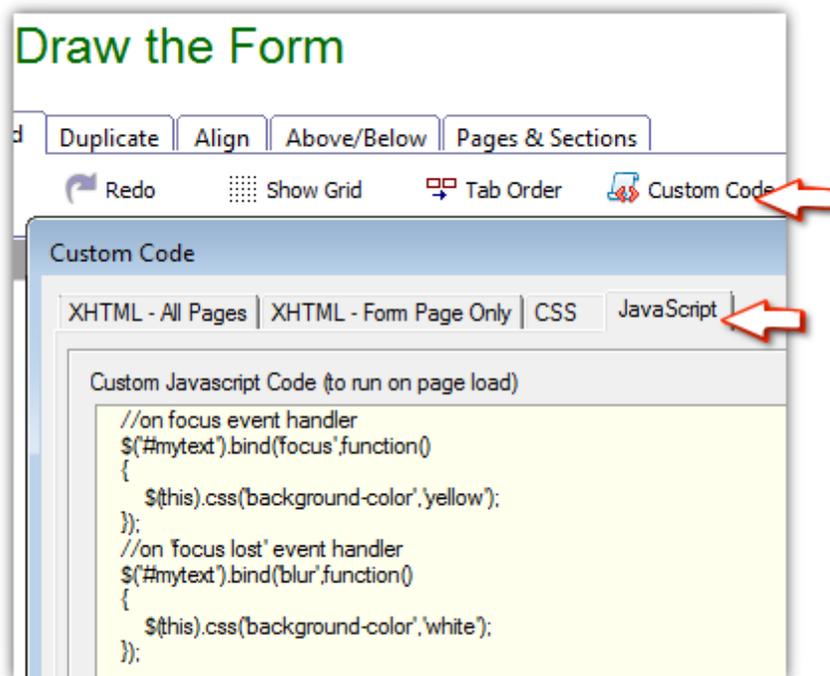
Note that Simfatic Forms itself will copy the custom scripts and style sheets to the output folder and will upload the files along with the rest of the code.

Javascript initialization code

Some custom widgets might require doing initialization on loading the page. (Like initializing the options, customizing the behaviour) You can put your custom Javascript initialization code here.

Custom code dialog box

You can customize the generated pages adding custom CSS, XHTML or JavaScript code. Press the 'custom code' button in the Draw the form page to open the custom code dialog box.



There are three tabs in the 'custom code' dialog box.

XHTML - All Pages

The code you provide in this tab will be applied to all generated pages (The form page, confirm page and 'thank you' page) So, if you want to apply a certain customization to all pages (like a common header and footer), provide it here.

Code to place in the <head> tag

This is the ideal place to include external style sheets or JavaScript files.

Header

The custom XHTML code to be placed on the very top of the page.

Footer

The code to be placed at the bottom of the page.

Note that you can provide custom JavaScript code in `<script>` and `</script>` tags

XHTML - Form page only

The code provided here is applied only to the form page. Provide the customizations specific to the form page in this tab.

If 'custom XHTML for all pages' is also provided, the code for all pages is applied first and the code only for form page is applied next.

CSS

The custom CSS code you provide here is added to the generated CSS file. If you want to modify or improve style of the form, provide your custom CSS code in this tab.

JavaScript

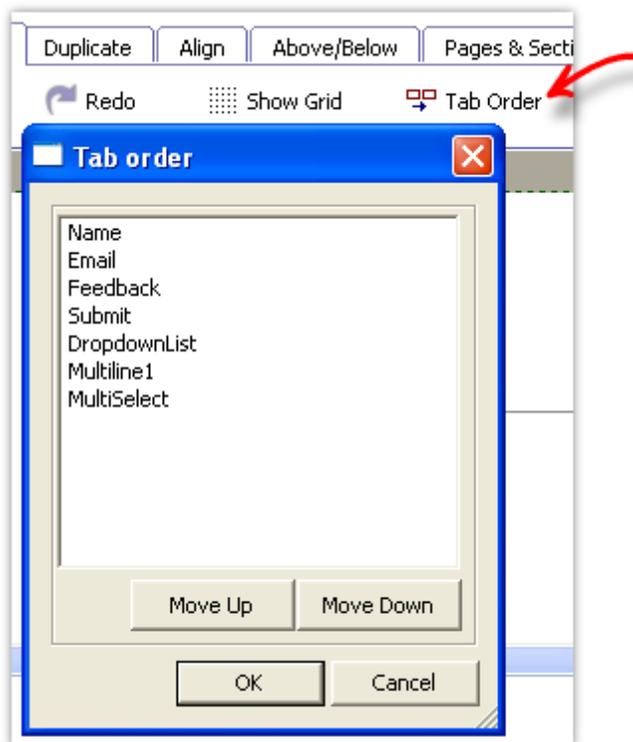
Put the custom Javascript code that should run on page load in this tab. You can use jQuery as well since most of the dynamic features (like validation) of Simfatic Forms uses jQuery

See Also: [How to handle JavaScript events in the form](#)

Tab order dialog box

In the tab order dialog box, you can change the tab sequence (The sequence in which the focus shifts from control to control when the 'tab' key is pressed).

Press the 'Tab order' button in the draw the form page to open the Tab order dialog box.



The tab order dialog box lists the input form elements in the form. You can change the sequence by moving the element up or down by pressing the corresponding buttons.

Move Up

Move the selected elements up in the tab sequence.

Move Down

Move the selected elements down in the tab sequence.

Page settings dialog box

Press the Page Settings button in the Draw the form page (Pages & sections tab) to open the Page settings dialog box.

Page Title

A descriptive name to the page. You can optionally display the page title (using the page title widget).

Display the page only in certain conditions

Choose this option if you want to display the page only on certain conditions. Press the 'Add new' button to add conditions

5.7 'Add Input Validations' page

In this page you can attach validations to the fields. The validations are meant to prevent your visitors from entering invalid or inconsistent data.

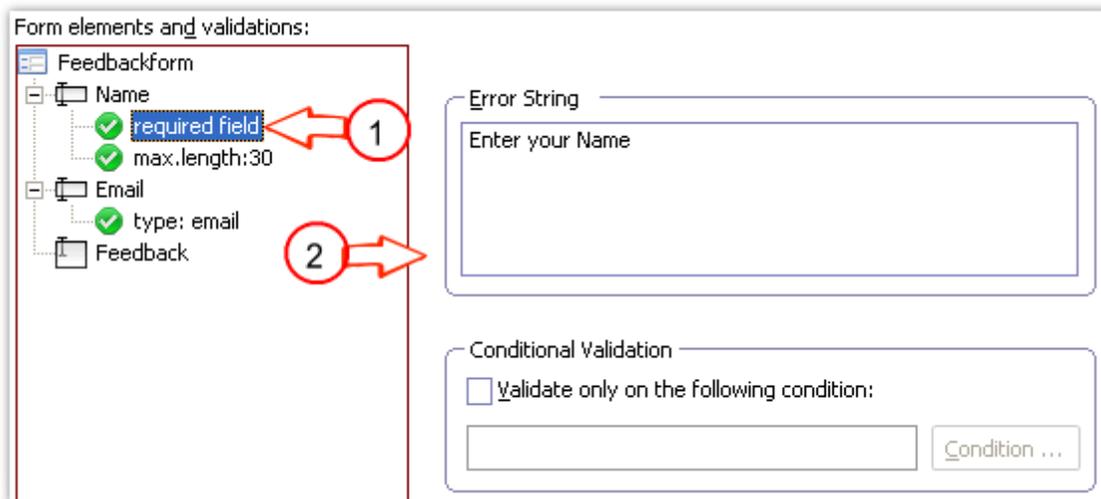
The layout of the page

The left side box shows the form shows the input fields and the validations in a tree structure.

Select the input element from the left side box and you can select the validations on the right side.

The validations that appear to the right side depend on the type of the element that you select on the left side.

Editing the properties of the validation



Select a validation node from the left side. The right side shows the properties of the validation.

Error String

This field contains the error displayed to your visitor when the validation fails.

See: How to edit the validation error messages

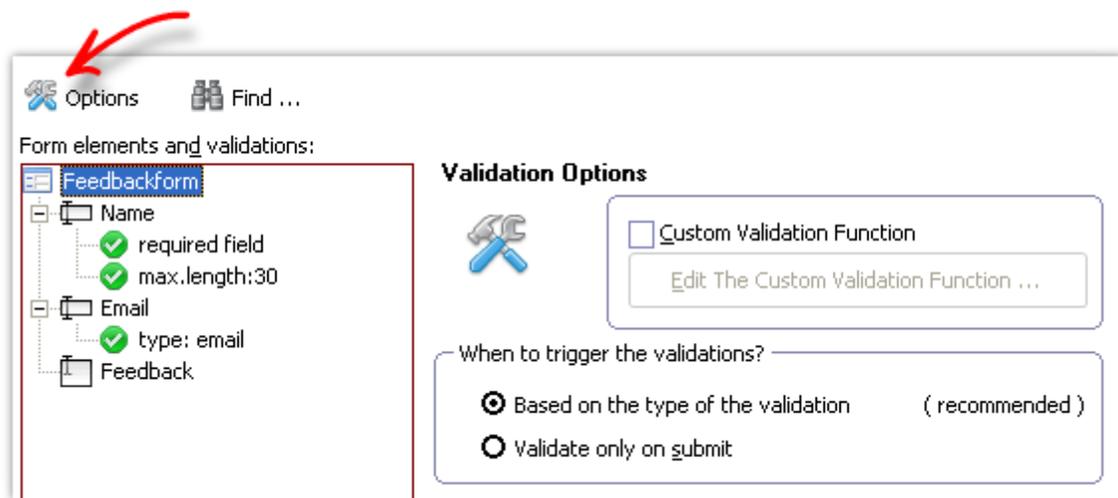
Conditional Validation

If you want to conduct the validation only under some conditions, then you can use the conditional validation feature. Click on the 'validate only on the following condition' check box Then click on the Condition button to open the Validation condition dialogbox

In the 'validation condition; dialog box, you can select the condition when the validation needs to be enabled

See: How to add a conditional validation

Validation Options



Press the options button to edit the general validation options.

Custom Validation Function

If the built-in validations do not meet your requirement, you can custom-code the

validation in the custom validation function.

The custom validation function has a single parameter: the form object.

If you have more than one validations, then you can ladder all those conditions in the same custom validation function.

If the validation fails, the function should return an object containing the element and the error message like this:

```
if (failed)
{
  return ({ elementobj:formobj.elements['userid'],
  message:'user ID not found!' });
}
```

See: [How to add your own validations](#)

When to trigger the validations

You can conduct the input validations based on the type of the validations or when the visitor submits the form. In the first method, the validation is triggered based on the type of the validation. For example, the numeric validation is triggered as the user types the value (so that the user gets immediate feedback). Similarly, email validation is done when the user switches the focus. This method focuses on giving the least intrusive feedback as early as possible.

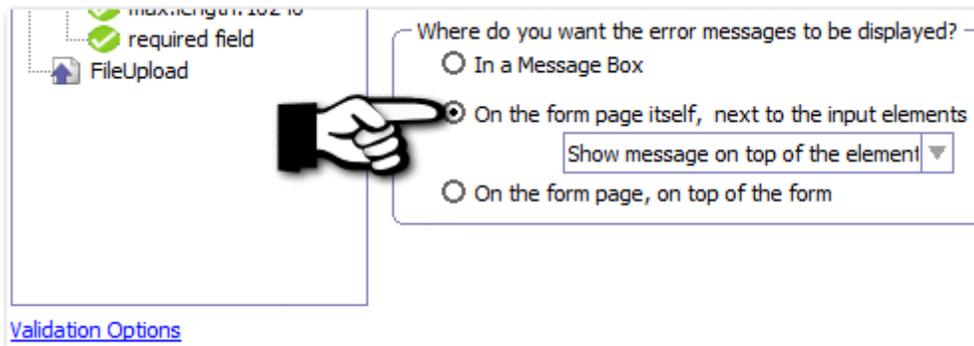
In the second case (validation on submit), the validation done only when the form is submitted.

Please note that in the first method, the form submission also triggers all the validations. It makes sure that form is submitted only after the validations.

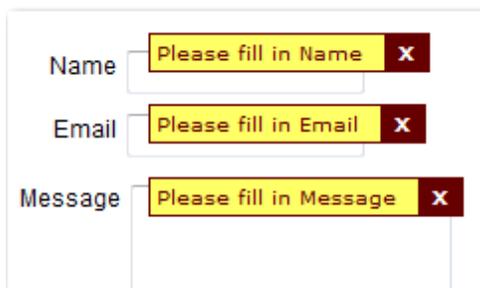
How do you want to display the error messages

You can have the error messages displayed in a message box or have the error messages displayed in the form page itself. The error message can be displayed in a floating error box or in a single box on the top of the form.

Floating error box style



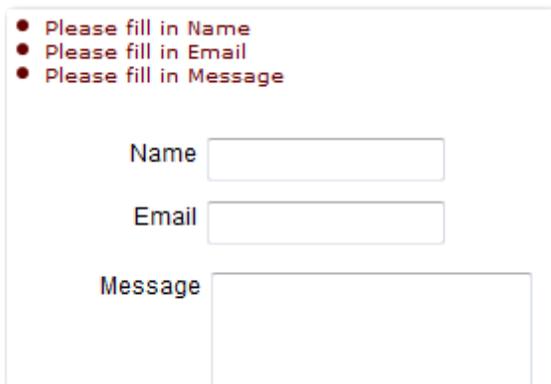
Selecting the type of error message display



Floating error box shown in the browser

Single error box on the top of the form

Select the 'On the form page, on top of the form' to display the error messages on top of the form.



A screenshot of a web form with three input fields: 'Name', 'Email', and 'Message'. Each field is empty. Above the fields, there are three red error messages, each preceded by a red dot: 'Please fill in Name', 'Please fill in Email', and 'Please fill in Message'.

Single box error shown in the browser

See Also:

Smart Validations

Unique validation requires Database

Validation Types

This section documents the different types of validations.

Textbox Validations

Depending on the input expected from your visitor, you can add different types of validations to the textbox.

When you select a textbox element from the left side element tree in the input validations page, a tabbed box containing the textbox validations are shown on the right side.

Basic Validations

Required Field

The visitor must provide input to this field

Input Type

Validate the data type of the field

The following types are supported:

- Alphabetic

Only English alphabetic characters allowed.

- Alpha-Numeric

English alphabetic characters and numeric characters allowed.

For both the types above, no spaces are allowed by default; if you want to let spaces, then select the 'Allow spaces' checkbox.

- Numeric

Only numbers allowed.(examples: 23.14 -1000 1000.00 1,000,000)

- Email

Validates the input for an email pattern

Size

You can limit the size of the input using the size validations. In order to enable a validation, select the checkbox next to it.

- Max. Length

The maximum number of characters that the input can contain.

- Min. Length

The minimum number of characters that the input should contain.

- Greater Than

The input value should be greater the given value. (The input type should be Numeric)

- Less Than

The input value should be less the given value. (The input type should be Numeric)

Comparisons

It is quite often required that you want to validate by comparing the values of the input fields.

You can add comparison validations in the 'Comparisons' tab. The left side in the comparison tab has a listbox that will contain all the comparison validations.

There are buttons for adding and removing the comparison validations in the right bottom side.

'Add' button

To add a comparison validation, press the Add button.

This opens the 'Add Comparison' dialog box Select the comparison and the other input field to compare it to. Press OK. You can add more than one comparison validations for the same input field.

Remove

This button is to remove a previously added comparison validation.

Select the comparison from the list and press the 'Remove' button.

Pattern

This validation could be used if you are expecting a fixed pattern in the input (like phone number, SSN etc)

In addition, the pattern validation could be used to limit the allowed/disallowed characters.

The pattern matching requires a coded string known as 'Regular expression'.

If you have the regular expression you can provide it directly in the 'regular expression' box.

For easily creating the regular expression patterns there are two methods:

Make Pattern

You can use a wizard if you want to limit the allowed/disallowed characters in the input. Click on the 'Make pattern' button to open the 'Make Pattern Wizard'

Select from List

Click the 'Select from list' button to open the list of ready-made regular expression patterns. If the pattern required is provided in the list, you can select it and press OK.

Checkbox (single) Validations

The following are the validations that you can apply for a single check box.

The visitor should select this checkbox

There will be a validation error if your visitor submits the form without checking this check box

The visitor should NOT select this checkbox

Your visitor should not leave this check box checked while submitting the form. This validation could be used in cases where the visitor has to 'Agree' or 'Decline' to some agreement.

For a given checkbox, you can select either of the above validations; not both.

Checkbox Group Validations

The following are the validations that you can apply for a checkbox group.

The visitor should select at least 'n' check boxes

The visitor should select the selected number of check boxes or more from the group.

If lesser number of check boxes are selected, then a validation error happens.

If you want to enable this validation, select the check box and then enter the minimum number of check boxes to be selected.

The visitor should select a particular checkbox

The visitor is required to select the checkbox with the given caption. If that checkbox is not selected, then a validation error occurs.

The visitor should NOT select a particular checkbox

The visitor is required leave the checkbox unchecked. If that checkbox is selected, then a validation error occurs.

Radio Group Validations

Required Field

The visitor should select an option from the Radio Group.

The visitor should select

A validation error happens if the visitor selects an option other than the particular one.

The visitor should select

A validation error happens if the visitor selects the particular option.

Dropdown List Validations

The listbox input has only one type of validation - the validation to make the listbox input mandatory.

"The visitor should not keep the following item selected"

You want to force the visitor select a valid option from the list. You can add an extra item (something like "Select an Option ...") and keep this item selected by default. Then add the validation "The visitor should not keep the following item selected" with item as the default "Select ..." option.

On submitting the form the user has to select an option other than the default "Select ..." option.

Multiline Validations

The following are the validations supported for multiline textbox.

Required field

The visitor should not leave the multiline text box empty; some value must be entered. Select the check box to enable the validation.

Maximum length

The permitted size of the input in number of characters.

Select the checkbox and then enter the maximum input length expected.

Minimum length

The minimum required size of the input in number of characters.

Select the checkbox and then enter the minimum input length expected.

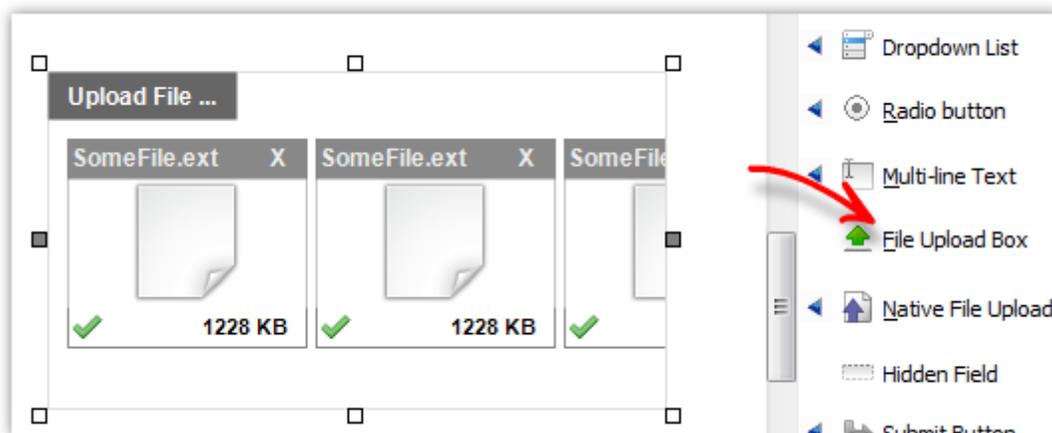
Allow/Disallow some characters

Press the 'Select' button to choose the characters that need to be allowed or disallowed. The 'Make pattern wizard' creates a regular expression pattern that allows/disallows the chosen set of characters.

File Upload Box Validations

The file upload box enables your visitor to upload files along with the form submission. It is important that you validate the type and size of the files that the visitor may sent to you.

The following types of validations are supported for file upload box



Required field

The visitor must upload a file.

Max. Number of Files

The File Uploadbox allows uploading more than one files in the same upload field. You can set a maximum limit in the validations.



The native File upload does not support uploading more than one files

Valid file extensions

Validate the file extension. You can separate more than one file extensions using comma(,).

For example, if you want to let only jpg or gif files, then provide Valid extensions value as jpg, gif

Max.Size

The maximum permitted file size

Check the Max size checkbox and provide the size. The Size could be in KBs or MBs.

Note: The 'Max.Size' validation is supported in server-side only.

That means, you will not get an error popup message box immediately after submission; but only after the server processed the input.

This is because the client side script (JavaScript) doesn't have access to the file system.

Date Picker Validations

Date picker allows selecting a date easily. You can have date comparison validations.

When you select a Date picker field from the left side element tree in the input validations page, a tabbed box containing the date picker validations are shown on the right side.

Required Field

The visitor must provide input to this field

Date comparisons

The date value should be before/after a chosen date.

You have the following options:

Same day

The date value is compared with the day on which the form is submitted. For example, if 'Should be before same day' is selected, and if the form is submitted on 9 April 2009, then a date value 1 Mar 2009 is valid whereas 10 April 2009 is not accepted.

A fixed date

Choose a fixed date for the comparison. The date input is compared with the selected fixed date.

Another date field

If you have another date field in the form, you can compare the date with the other date field as well.

Doing more validations on the date field

It may be required that you want to do validations after finding the number of days between two dates. You can use a calculation field in this case. The calculation field supports functions like `days_from_today()` and `days_between()`. The calculation field supports comparison validations like greater than and less than. The calculation field can be made hidden as well.

For example, imagine that you have a date of birth field in your form. You want to limit the age above 15.

You can add a calculation field (named 'age') with a formula `age(date_of_birth)`. Now, you can add a validation for the calculation field to be `age > 15`.

Number Field Validations

The number field accepts decimal numbers only. You can have additional validations on the number entered.

When you select a decimal field from the left side element tree in the 'add input validations' page, a tabbed box containing the number field validations are shown on the right side.

Basic Validations

Required Field

The visitor must provide input to this field

Size

You can limit the range of the entered number using the range validations.

- Greater Than

The input value should be greater the given value.

- **Less Than**

The input value should be less the given value.

Comparisons

It is quite often required that you want to validate by comparing the values of the input fields.

You can add comparison validations in the 'Comparisons' tab. The left side in the comparison tab has a listbox that will contain all the comparison validations.

There are buttons for adding and removing the comparison validations in the right bottom side.

'Add' button

To add a comparison validation, press the Add button.

This opens the 'Add Comparison' dialog box Select the comparison and the other input field to compare it to. Press OK. You can add more than one comparison validations for the same input field.

Remove

This button is to remove a previously added comparison validation.

Select the comparison from the list and press the 'Remove' button.

Calculation Field Validations

The calculation field validations are same as Number Field Validations

Multi-select List Validations

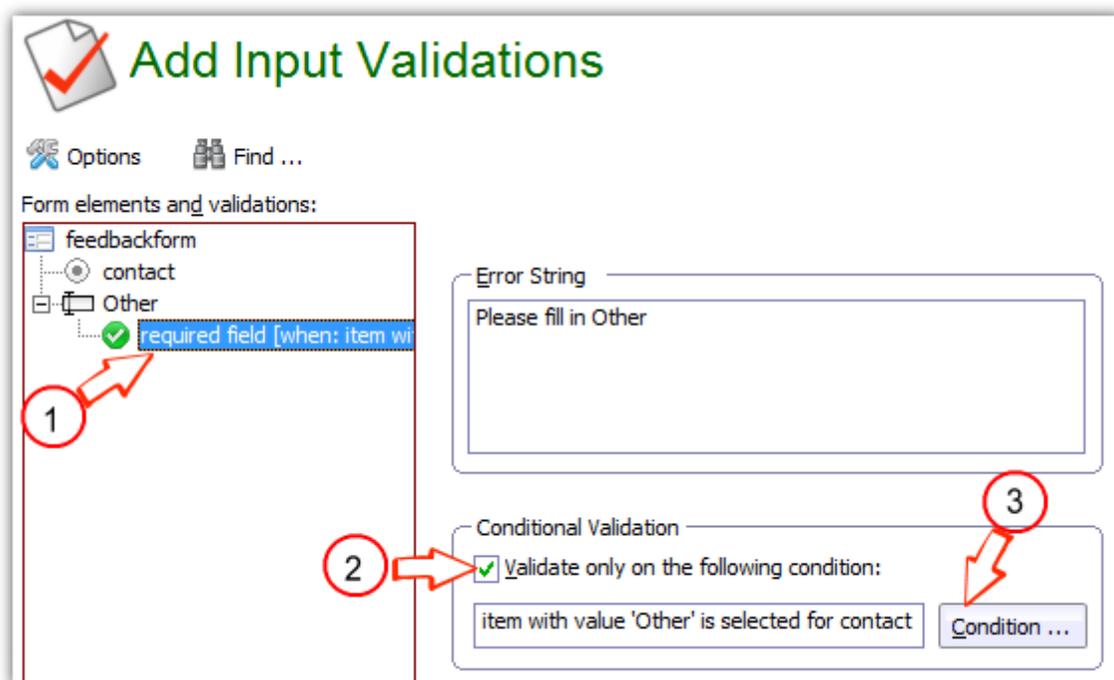
The multi-select list has same validations as the Checkbox Group

Native Fileupload Validations

The native file upload validations are same as the File Upload Box Validations

Validation condition dialogbox

If you want to enable a validation only under a condition, you can use the conditional validation feature.



Select the validation node (in the Add Validations Page) and then select the Validate only on the following condition checkbox. Click on the 'Condition' button. The Validation Condition dialog opens.

First select the Form Element whose value you want to test. The conditions vary depending on the type of the element.

For Text box elements, you can select from 'is', 'contains', 'greater than' and 'less than' conditions.

For example, do the validation when TextValue 'is' 'yes'

For checkboxes,radio buttons and list boxes, the condition is whether or not a particular item is selected.

5.8 Form Processing Options Page

You can configure general server-side form processing options in this page.

Show a Confirm Data Page

A confirmation page adds an additional step before the actual submission. If this option is enabled, the confirmation page is shown when the form is submitted. The visitor has to verify the form data in the confirmation page and click on the 'Confirm' button for the final submission. Note that the form submission is not complete until the user presses the 'Confirm' button.

Enable this option only if you want a two-step submission.

See Also: [Edit Confirmation Page](#)

Send Form submission emails

Enables the option to send the submitted form data to one more email addresses.

You can customize the form mail options (the email addresses to send the form data, the email template etc) in the [Configure Form-mail Page](#)

Send an auto response to the visitor who submitted the form

You can send an auto-response to the visitor who submitted the form.

The form need to have a field to accept the visitors email address and preferably, a field for her name.

You can configure the additional options for the auto-response email in the 'Configure Auto-response' page.

Save the form submission data in a database table

The form submission data is saved in to a database table. Only MySQL database is supported in the current version. See: 'Save to database' page

Save the data from the form submissions to a file on the server

The data from the form submissions are saved to a file in the server. The data is saved as a CSV (comma separated values) file. The Form to File Page gives additional options.

Do payment processing using PayPal

You can accept payments from your forms through PayPal. Check the option and configure your Paypal account in the Setup PayPal Page.

Do custom processing using your own Extension Modules

You can add your own custom functionality to the form by using Extension Modules. Choose this option to enable Extension Modules.

See : 'Extension Modules' Page

General Settings

This section contains general form processing settings.

Default Time Zone

The time zone to be used for date and time. The fields like form submission time uses this setting.

Please note that the *date input fields* in your form is not affected by this setting.

Email Format

Determines the format of the emails sent (form submission email, auto-response email etc). Could be text mail (no formatting) or HTML formatted emails.

Allow loading form values from URL

If this option is enabled, you can initialize the field values through the URL of the form.

For example, suppose you have fields *Name* and *Email* in your form. You can open the form loaded with values John and john@hiswebsite.com by opening the URL:

```
http://yourwebsite.com/forms/signup/signup.php?  
Name=John&Email=john@hiswebsite.com
```

Troubleshooting options

Enable Logging

If logging is enabled, the script logs errors and other general information to a text file on the server. In case any error happens, you can download the log file to check what went wrong.

See Also:

- Download log file dialog box
- Turn logging off for better performance

Email address to receive errors

If an email address is provided in this field, then the script sends errors to the provided email address. This is very helpful because you will come to know of the errors immediately.

It is recommended that you provide an email address in this field for every form that

you make.

Debug mode

Display errors on the page itself. This option helps getting the errors immediately on the screen.

If debug mode is unchecked, the errors are not shown on the screen (so that your website visitors won't come to know about the details of the error)

Enable Security validations to prevent spammers

When this option is enabled, the script will do special checks to prevent automated form submissions. If this option is enabled, you won't need Captcha in the form to protect the form from spammers.

It is rarely that you will need to uncheck this option.

Reject submissions from selected IP addresses

The form submission email contains the IP address of the form submitter. In case you observe that some particular user keeps sending you form submissions in an attempt to spam you, you can ban his IP address.

5.9 Configure Form-mail Page

The form data submitted in the form could be directed to one or more email addresses. You can configure the form-mail options in this page. This page is shown only if you enable form-mail in the Form Processing Options Page.

The fields in this page

Recipients' field

The email addresses to which the form submission mail should be sent. Press the 'Select ...' button to select the email addresses. The Address book opens. You can choose the email addresses from the list of addresses.

Reply-To

The reply-to field in the email is useful when you try to reply to the form submission email. When you reply the email from the email client, the address in the 'Reply-To' field appears in the 'To' field.

You have two choices.

1. A 'fixed' email address to be used in the 'Reply-To' field of the email
2. A field in the form can be used as the 'Reply-To' field .

If you have an email field in your form where the visitor enters her email address, then select the email field in the second option.

The second option is helpful for replying to the form mail quickly. You can just press the 'Reply' button in your email software to reply to the form mail.

Press the 'Select' button to open the The 'Reply-To' field dialog box

Subject

Enter the subject of the form submission email.

Email Content

An email template is used to send the form data.

You can format the template to include the field values from the form. The field value place holders are represented by the notation `%field_name%`. While processing the form, these place holders are replaced by the actual values from the form before sending the email

By default, an email template is generated and placed as email content.

'Auto generate' button

Click on the 'auto generate' button to generate a simple email template. The generated email template will contain the form field names and the place holders for the field values.

Pressing the 'Auto-generate' button will show the Insert Field Table dialog box. Here, you can edit the field table that will be inserted in the generated template.

Insert Field

Insert the place holder for the selected field. For example, if you insert the place holder for a field 'Name' then it is represented as `%name%`. While processing the

template, the %name% will be replaced with the actual value submitted in the form.

Insert Field Table

This button pops up the Insert Field Table dialog box. You can edit the field table and insert it in the template.

Attach file uploads to the email

This checkbox appears if there are file upload boxes in the form. If the box is checked, the form mail will have the uploaded files attached.

Editor

The editor button is displayed only if you have enabled HTML emails in the Form Processing Options Page page.

Click on the editor button to open the HTML editor where you can edit the email template. The HTML editor window has more formatting features.

Using the email template: An Example

Let us see an example how an email template is processed.

Suppose you have given the following text as the email template

```
You have a good news from your form processor.  
A visitor has submitted the form data:
```

```
Name : %name%
```

```
Email: %email%
```

Suppose somebody submits name 'John' and email 'john@hisdomain.com'
You will receive the email like the one shown below:

```
You have a good news from your form processor.  
A visitor has submitted the form data:
```

```
Name : John
```

```
Email: john@hisdomain.com
```



You can have field place holders in the subject field also.

The 'Reply-To' field dialog box

The 'Reply-To' field dialog box allows you to choose the 'Reply-To' field of the form mail. You can either select a 'fixed' field or choose to have the value of a field from the form itself.

If your form has a field to collect the email address of your visitor, then select the field as the 'Reply-To' field. This helps in replying to the form mail quickly. When you receive an email from a form submission, you can just press the 'Reply' button in your email software to reply to the visitor who submitted the form.

5.10 'Edit Confirmation Page Template' page

Confirmation page introduces an extra step in to the form processing. When the visitor submits the form, a page is displayed, that contains the submitted form data and a 'confirm' button and an edit button.

The visitor has to press the 'confirm' button for making the final submission. If required, the visitor can press the 'edit' button to make changes to the submission.

You can edit and customize the confirmation template in the 'Edit Confirmation Page Template' page.

The fields in the page

You can edit the confirmation page template in the editor in the middle of the page. The values of the fields in the form are represented using the notation %field_name%. While processing, these field value place holders are replaced by the values of the corresponding fields.

Auto Generate Page

Automatically generate a simple 'confirm data' page template. The automatically generated page will contain a table of all the field names and the field value place holders plus a confirmation button place holder.

'Insert Field' button

Insert the place holder for the value of the selected field. While processing the form, the place holder will be replaced by the value of the corresponding field.

'Field Table' button

Opens the Insert Field Table box where you can select and re-arrange the fields in the

table. You can as well choose to show only the non-blank fields.

'Confirm' button label

The label that should be given to the 'Confirm' button. You can change the default label with the one you like. Or, you can give the 'Confirm' label translated in to your language.

'Edit' button label

The label on the 'Edit' button - the button to make changes to the form submission.

Button type

You can have customized image buttons or the standard buttons in the Confirmation page. The image button style could be changed in the Look and feel page (under the 'Image button' style)

'Editor' button

Open the HTML editor Window that has more editing space and more formatting options.

Load from HTML file

Press the 'Load from HTML file' button to load a pre-created HTML page.

You can create the HTML page in your favorite editor and load it here. Then you can press the 'Insert All Fields' button to place the fields in the page.

5.11 'Auto-response' page

Auto-response is the email sent back to the visitor who submitted the form.

In order to use this feature, there should be two fields in your form : the fields for the name and the email of the visitor.

The fields in this page

Name Field

The input field where the visitor submits her name

Email Field

The field where the visitor submits her email address.

Subject

The subject of the auto-response email.

Email Content

The 'body' of the auto-response email. You can include the field values from the form submission data in the body of the email. The field value place holder is represented by the notation %field_name%

Insert Field

Inserts the place holder for the currently selected input field value at the current insert position.

Editor

The 'Editor' button appears only if you selected the format of the emails as HTML. Press the 'Editor' button to open the HTML editor window that provides more features for formatting the email.

An example

Let us assume that your form is a service enquiry form with fields 'name', 'email' and the type of service required.

Suppose the auto response email template given is

```
Dear %Name%
```

```
Thanks for submitting the enquiry. Our representatives  
will contact you soon regarding our %service% service.
```

```
Thanks  
OurCompany
```

Suppose your visitor submits the form with the following information: name: John
email: john@hisdomain.com
service: resume writing

Our dear John will soon receive an email like the following:

```
Dear John
```

```
Thanks for submitting the enquiry. Our representatives  
will contact you soon regarding our resume writing  
service.
```

Thanks

OurCompany



You can have field place holders in the **subject** field also.

For example: Thanks for the submission, %Name%

See Also: [How to send an auto response](#)

5.12 'Form to File' Page

If you have enabled the feature to save the submitted form data to a file in the server, this page is displayed. The form data is saved in a file as coma separated values (known as CSV file). The first line is the column titles. The data submitted in each form submission is appended after the first line.

See Also: How to save form submissions to a file

The fields in this page

There are two list boxes in this page. The right side list box contains the fields to be saved to the file.

The left side list box contains the fields that are NOT to be saved.

By default, all likely fields are added to the 'to be saved' list.

The fields are saved in the file in the order they are present in the list box. That means the field in the top of the list appears left most in the CSV file and the bottom most field appears in the right end of the CSV file.

'Add' button

Select the item to be added from the left side list box and press 'Add' button. The selected field is added to the 'to be saved' list in the right side.

You can select more than one field from the left and press the 'Add' button to add them together.

'Remove' button

Select the fields to be removed from the right side box and press 'Remove' button to remove them from the 'to be saved' list.

The removed fields appear in the left side box.

You can select more than one field from the right side box and press the 'Remove' button to remove them together.

'Move Up' button

Select the field to be moved up in the right side box and press 'Move Up' button to move the field up. The order of the fields in the right side box affects the order of the fields in the form data (CSV) file.

'Move Down' button

Select the field to be moved down in the right side box and press 'Move Down' button to move the field down.

File size limit

As your visitors keep on submitting the form, the form data file grows. You can set the file size limit so that when the file size reaches the specified limit, the data till then is backed up to a file and marked (renamed) with the date. A new file is created where appending the form data continues. You can download the old file and delete it from the server to save space.

Enter the value of the 'File Size limit' in KBs (Kilo Bytes).

The default value of the file size limit is 1024 KB (1 MB).

5.13 'Save to database' page

You can have the form submission data saved to a table in a database. (The current version supports MySQL database only). Choose the 'save form submission to a database table' option in the form processing options page.

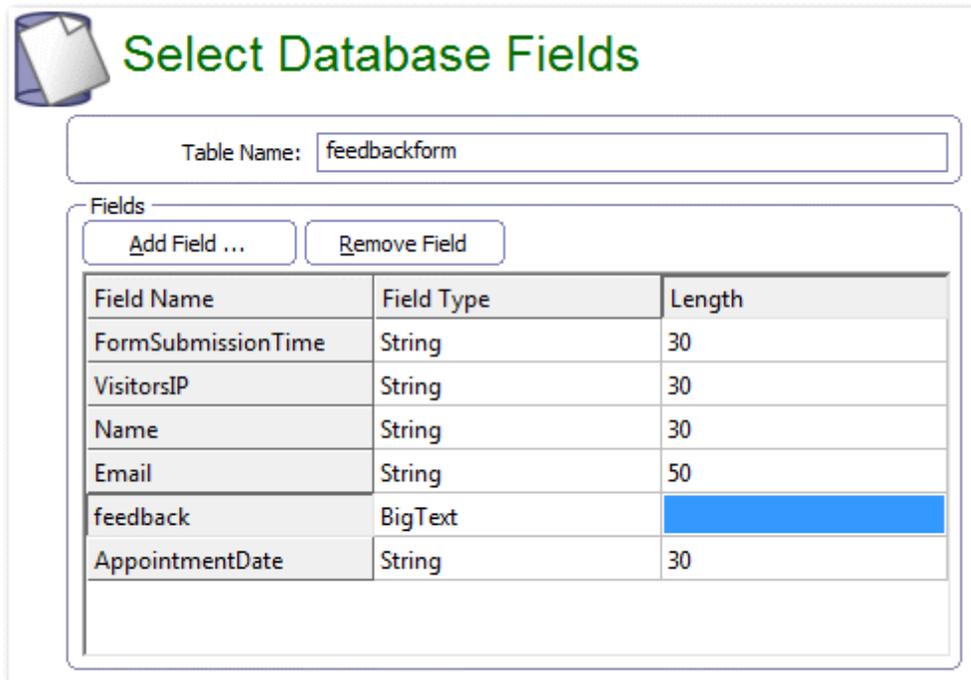


Table Name:

Fields

Field Name	Field Type	Length
FormSubmissionTime	String	30
VisitorsIP	String	30
Name	String	30
Email	String	50
feedback	BigText	
AppointmentDate	String	30

Table name

The name of the table where the form submission data is saved. By default, the name of the form is provided as the table name. The form processor script will **create** the table in the database.

If the table is already existing, the script will **alter** the table to match the field description that you provide in the 'Save to database' page.

Fields

The fields in the table. The fields list is pre-populated with the fields in the form. In addition to the fields that you added to the form, two special fields - the form submission time and the IP address of the visitor who submitted the form- are also added to the field list.

Press the Add Field button to add a field that is not already present.

Select a field and press the 'Remove Field' button to remove a field.

You should update the field type and the field size according to the needs. It is also recommended to enforce size and type checks by having appropriate form validations as well.

5.14 Setup PayPal Page

You can setup PayPal Website Payments Standard to accept payments from your form users.

First, select 'Do payment processing using PayPal' option in the Form Processing Options Page

In the PayPal Setup page, press the 'Configure PayPal' button to start configuring PayPal payments.

Setting up PayPal

It is recommended to use the PayPal sandbox first to try the PayPal processing. Get a developer account and setup some test accounts.

Once you have tried the payment processing, Get a PayPal account and move to production mode.

PayPal General Settings:

The Email Address is the one that you have signed up with PayPal. If you are trying the test mode, provide the email address from one of the test accounts.

Check the Test mode option to first try the PayPal integration. Then later, you can uncheck the option and re-upload the form to move to production mode.

For one time payments, select product/service. For recurring payments, choose 'Subscription'.

Products page:

The item name, unit price and quantity can be directly given in the corresponding boxes or can be from the form fields. To select a form field, press the right arrow button and choose the field.

You can add more than one product as well. Press the 'Add more items' button and provide the details.

Additional Charges

The additional charges and discounts fields are optional. You can provide the values or select from the form fields (like calculation fields).

Subscription Setup

If you have selected 'Subscription' in the first page, you can setup the subscription details in this page. Select the Billing cycle and the amount. You can setup a trial period too.

Donation Setup

If the payment type is 'Donation', you can provide a description and amount in this page.

Conditionally send to PayPal

In certain cases, you may want to send the user to Paypal only on certain conditions. (like user selecting an option to 'make the payment now'). You can make the PayPal step conditional by selecting 'Send to Paypal only on certain conditions' check box.



If Paypal is enabled for your form, the rest of the form processing steps (send form submission email, auto response, save to database etc) will be executed only after the successfully completes the payment at Paypal.

5.15 'Extension Modules' Page

Extension modules allow you to add your own custom processing to the form. For example, if you want to submit the form submission data to a custom web service, an Extension Module can be created and plugged in to your form.

See the Extension Modules page for more information on how to code extension modules.

To enable Extension Modules, enable the 'Do custom processing using your own Extension Modules' option in the Form Processing Options Page.

You can have any number of extension modules attached to a form. The 'Extension Modules' list will contain all the extension modules that you have added to the form.

Add Module

Press the 'Add Module' button to add a new extension module to the extension modules list.

Remove Module

Select a module from the list and press the 'Remove Module' button to remove a module from the list

Remove All

Pressing the 'Remove All' button removes all the modules from the modules list.

5.16 'Thank You!' Page

'Thank you' page is displayed when there was no error and the form was processed successfully.

Once the form is processed successfully, you have two options:

- you can redirect to a URL
- you can display a 'thank you' page

Redirect to a URL

Select the 'URL' radio button and enter the URL to which to redirect. The URL should be absolute URL (like <http://www.my-website.com/folder/page.html>).

Edit the 'thank you' page

You can create the content for the 'thank you' page. Select the 'HTML Page' option.

'Auto generate' button

Generate a simple thank you page. You can edit and customize the generated page. This is an option that you can use if you are in a hurry!

'Edit' button

Opens the HTML editor. The HTML editor window has more features for formatting the page.

'Load File' button

Load the content from an existing HTML file. This option could be used if you already have a page/template created.

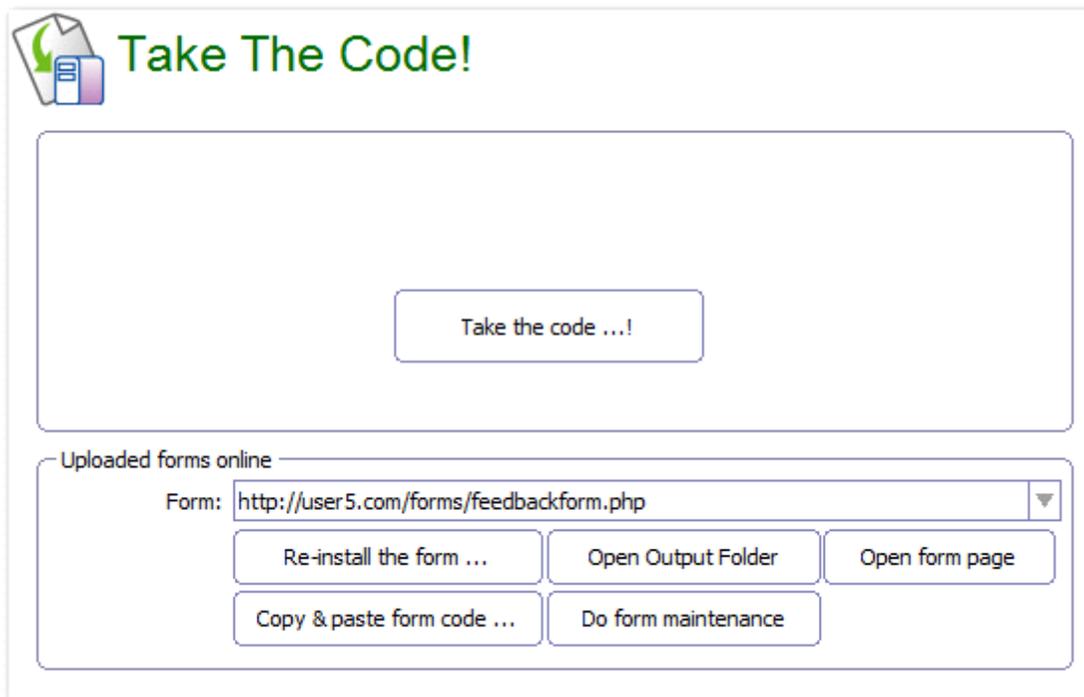
'Insert field' button

Select a form field from the drop down list and then press the 'Insert field' button to insert that field into the 'Thank You' page template. The field placeholders will be replaced with the actual values submitted in the form.

 If you use non-English characters in the page, use UTF-8 encoding. This will make the character display properly.

5.17 Take the code page

In this page, you can generate the code for the form and upload the script to your web server. The take the code wizard takes you through the steps. Press the Take the code button to start the process.



'Take the code' button

Opens Take the code Wizard that takes you through the process of generating the form code and installing the scripts on the server.

Uploaded forms online

This set of options appear only after you have successfully installed the form on a web server.

The operations in this section are for a particular installation of the form (You can install the same form on multiple websites or multiple locations on the same web server.)

Choose a URL from the list to work on that particular installation.

Re-install the form

Install the form again on the same server. Use this when you have made some changes in the form and want to re-upload the form.

This option uploads only the changed files. If you want to re-upload all the files, choose 'fresh install' from the menu.

Open Output folder

Open the form page in browser. You can quickly test your form using this option.

Open form page

Open the form page in browser. You can quickly test your form using this option.

Copy & paste form code

Press the copy & paste form code button to open 'Copy and Paste Form Code' dialog box. You can copy the form code that you can paste in your web pages.

Do form maintenance

Takes you to the 'Form installation maintenance' page where you can view the form submissions, download form submission files and more.

Take the code Wizard

The take the code wizard takes you through the steps for generating the code and uploading the script to your web server. Press the 'Take the code' button (in the 'Take the code' page) to start the process.

See Also

How to install the form on a subdomain

General page

Provide the name of your website where you plan to install the form. You can either provide a new website name or pick an existing name. If you pick an existing name, then the settings used for that web site will be preloaded in the wizard pages. This is handy when you install many forms on the same website.

Email settings page

Provide the advanced email configuration for the website. For example, you can provide the SMTP login details for sending emails. Select the 'Use SMTP server for sending emails' checkbox, press the 'SMTP settings' button and provide the SMTP server login details.

Database login page

If you have opted to save the form submission data in a database table, then this Database login page is displayed. Provide the database login details and press Next.

Generate code page

You can change the output folder if required. (A folder for the form will be selected

by default) Press the 'Generate code' button to generate the code. Press 'Open output folder' to open the folder where the code is placed. Pressing the 'Advanced options' button opens the Code Generation Settings dialog box.

Upload method

You can either use Simfatic forms' in-built FTP client to upload the form scripts to the web server or you can manually upload the scripts. Simfatic Forms' 'Swift Upload' is easier and is recommended.

Swift upload

This page is displayed if you select Simfatic Forms' swift upload as the upload method.

You will need to 'Add' your web server before you can upload your form. Adding the server is the process where you configure the server with Simfatic Forms. Press the 'Add Server' button to start the The Add Server Wizard that takes you through the process.

Select the server from the drop down, choose the remote folder to install the form and press 'Upload' button to start the upload process.

Manual Upload

Follow the steps in the 'manual upload steps' box to upload the form scripts to your web server. After completing the process, provide the URL of the form processor script in the 'Form processor script URL' box and press 'Next'.

See Also:

How to upload the form manually

Put the form in your web pages

This page provides the copy & paste form code. You can copy the form code to your web pages. You can do copy & paste later as well, using the 'Copy & paste form code' button in the take the code page.

SMTP settings dialog box

The SMTP settings dialog is for configuring an SMTP server with Simfatic Forms.

In the 'Take the Code' wizard -> 'Email settings' page, under Advanced Settings, there is an 'SMTP settings' button. Enable 'Use SMTP server...' option and press the SMTP settings button to open SMTP settings dialog box.

Please note that you need to use SMTP option only if you want to use a specific SMTP server to send emails. If PHP is configured to send emails, you can use it and there is no need to configure the SMTP setting.

Fields in the dialog box

SMTP host Name

The name of the SMTP server.

Port

The port configured for the SMTP server.

Use Authentication.

Enable this option if your SMTP server requires login name and password. After enabling authentication, provide your user name and password.

Press the OK button to save the changes

'Copy and Paste Form Code' dialog box

'Copy & paste form code' dialog box allows you to integrate the form with your web site. You can copy and paste the required code from the dialog box to your web pages. Press the 'Copy & paste form code' button in the 'Take the code' page to open the dialog box.

Embed the form

Embedding the form seamlessly integrates the form with your website. There are two options: you can integrate using iframe or using JavaScript. Press the 'show a sample page' button to instantly open a sample page integrated with your form. Also, see the source of the sample page (do View->source in your browser) to see how the form code is copied to the sample page.

Using JavaScript

This is the recommended method to embed the form. The JavaScript embed the form code seamlessly integrates your form to a web page.

IFrame

An iframe is created in your webpage and the form page is loaded in to the iframe. The form, the confirmation page and the 'Thank you' page are all displayed within your website page itself. This gives very close integration with your website.

Popup Form

If you want to show the form as a pop up, copy the code from this section. Press the 'show a sample page' button to check out the form in a popup.

Lightbox style popup

This code opens a simple popup with the form in it. The size of the popup by default is based on the largest page on your form. If your form is big, scrollbars will appear in the popup.

Window popup

If you want to open a standard window popup (where a new browser window is opened) use this code.



IFrame is not available in XHTML 1.0 or HTML 4 strict. If you want to keep your pages conformant with these standards, use the JavaScript Embed the form code.



The 'allowtransparency' attribute also is not supported by the XHTML/HTML specs. If you don't want transparency to the IFrame, you can remove the allowtransparency attribute.

Code Generation Settings dialog box

The code generation settings page allows you to customize the way the code is generated.

In the Take the code wizard->Generate code step, press the 'Advanced options' button to open the code generation options dialog box.

The 'code generation' tab

The options in the 'code generation' tab determine the way the code is generated. You can have all the code 'amalgamated' in to one single script. Some may prefer this option since the output contains only one script.(except images; image files will be separate in all cases).

Separate files for different file types

Each type of file (CSS, JS, PHP scripts is generated separately. The different types of files are placed in separate subfolders. You can change the sub-folder names in the 'files & folders' tab.

Separate Template files

You can have all the template files in single file or in multiple files. Template files include the template for the form mail, the auto-response mail, the form page itself and more. If you intend to modify the generated code, having separate template files will be better.

Single amalgamated script

All text files are combined into one script. Even the CSS, JS and the HTML template files are embedded within the PHP script.

The files & folders tab

You can change the names of the folders or the main form processor script file. Different types of files are placed in separate sub-folders. You can change the name of the folder in this tab.

The Add Server Wizard

When using Simfatic Forms to upload to webserver, you need to configure your server in Simfatic Forms with the FTP login details. The Add Server Wizard takes you step-by-step in to configuring a web server with Simfatic Forms.

In the Take the code wizard->Swift upload step press the 'Add Server' button to open the 'Add Server Wizard'

See Also:

How to install the form on a subdomain

The steps are explained below:

The Server Info page

In this page, you have to provide the basic information regarding the web server.

Web Address of the server:

The domain name of the website (like www.yourwebsite.com)

How do you publish web pages?

Select FTP if the server is remote and you used to upload web pages via FTP

Select Copy to folder if your web server is located locally and you publish web pages by copying the files to a local/network folder.

The FTP Login details page

This page appears only if you have selected FTP as the upload method in the first

step.

Provide the FTP login details of your web server in this page.

The fields in this dialog box are explained below:

Server address

The name of the server (like `www.myserver.com`) or the IP address of your server.

User Name

The user name to be used to login to the server.

Password

The password for FTP login.

Remember Password

Select this option if you want Simfatic Forms to remember the password. For the next login you don't have to give the password again.

The Publish folder page('Copy to Local Folder' case)

This page appears only if you have selected 'Copy to Local Folder' as the upload method in the first step.

The publish folder

Select the folder where you publish web pages. You have to select the folder where if we put a file say, 'page.html' it will be accessible as `my-web-site/page.html` where `my-web-site` is the address of your web server.

The 'Test' button

When you press the Test button, Simfatic forms copies a file to the publish folder and tries to access the page using the URL. This ensures that the publish folder you have entered is right.

The 'Publish Folder' page (in case of FTP Upload)

This page is displayed only if you selected FTP as the upload method in the first step.

In this page, you select the folder on the server where you publish your web pages.

If your web site address is `http://my-web-server.com` and if you want to publish a web page `page.html` such that its URL is `http://my-web-server.com/page.html`, then in which folder would you put the file `page.html`? Select that folder.

Folders

The folders box shows all the folders that are present on the server. You have to select the folder in which you publish your web pages.

The 'Test' button

When you press the 'Test' button, Simfatic forms uploads a file to the selected folder and tries to access the page using the URL.

The Summary Page

A summary of the choices you made are shown in this page.

Server Profile Name

Server profile name is a name that you can use to identify this particular web server configuration. The server profile name is displayed in the Install page combo box to select a server to install the form.

My Servers Dialog Box

This dialog box allows you to manage your web server configurations. The left side box in the dialog box shows the list of servers you have configured. Select one from the left-side box; the details appear on the right side.

Add

Configures a new server with Simfatic Forms. Click on the button to open the The Add Server Wizard

Modify

Modify the details of the currently selected server. Opens 'Modify server ' dialog box.

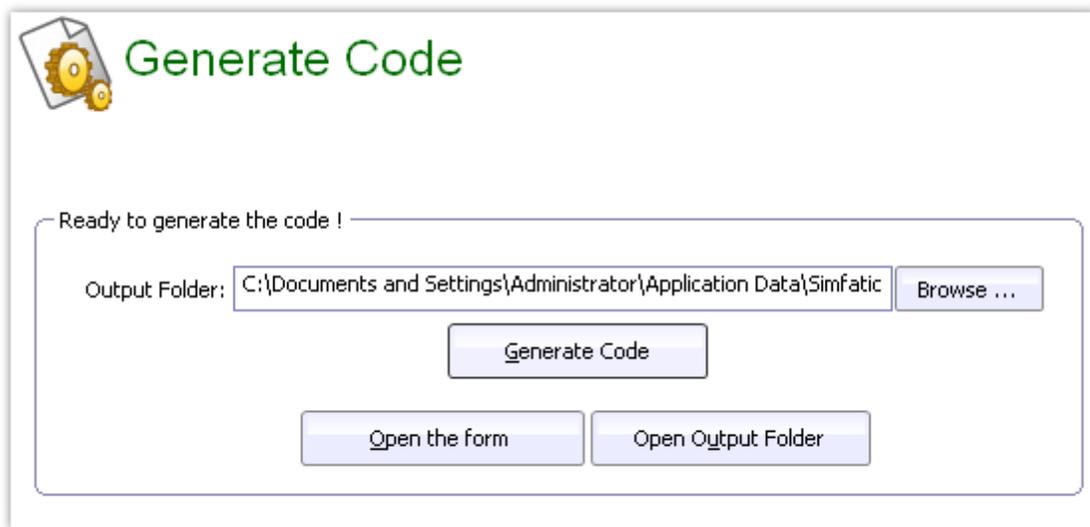
Delete

Deletes the currently selected server information from the list.

5.18 Generate code page ('client side only' case)

The Generate code page is displayed only in the case where 'generate client side code only' option is selected in General Form Properties Page

This page is to generate the HTML form (with the other client side parts: CSS and Javascript)



Ready to generate the code !

Output Folder: C:\Documents and Settings\Administrator\Application Data\Simfatic Browse ...

Generate Code

Open the form Open Output Folder

You can change the output folder. Press the Generate code button to generate the HTML form page.

Press the 'Open the Form' button to open the form in your browser.

If you want to edit/view the code, press the 'Open Output Folder' button.

See Also:

How does an HTML Form work?

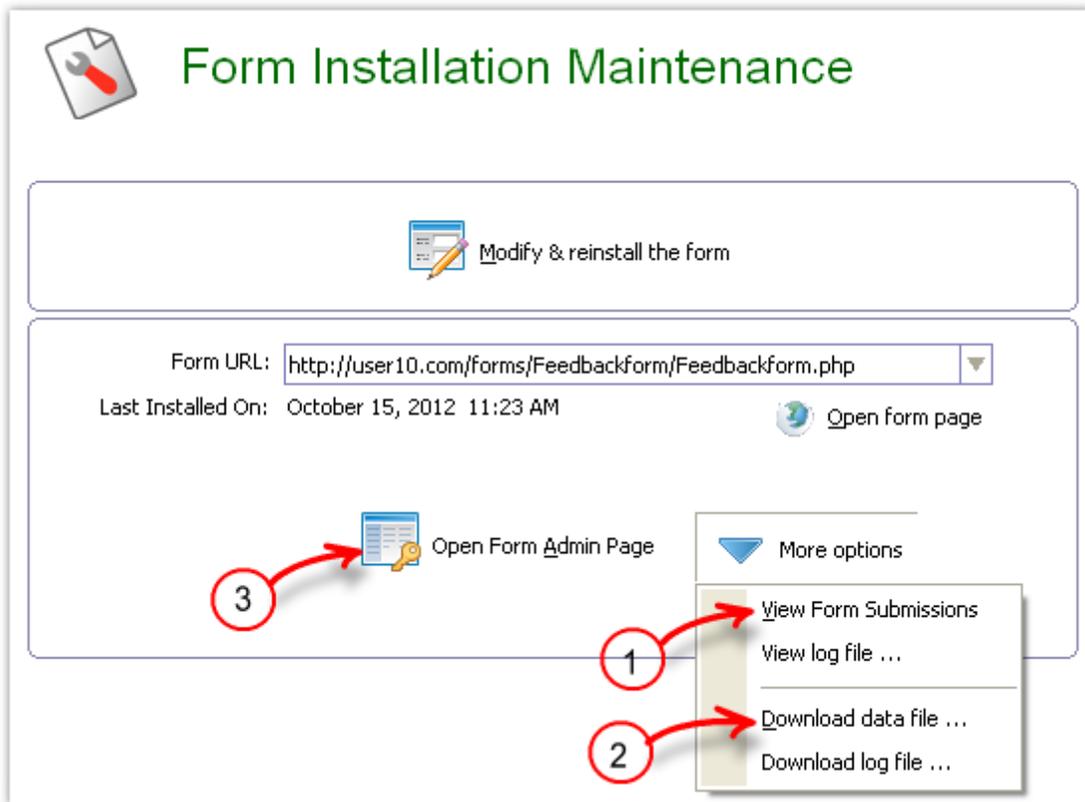
Why do I need to upload form to a webserver ?

5.19 Form installation maintenance page

Once the form is installed on the server, there are a set of operations (like view form submissions) that you may wish to do. The 'Form installation maintenance' page provides the options to maintain a form that was installed by Simfatic Forms on a web server.

If you are in the 'Form Edit' mode, you can switch to the Form Maintenance mode by

clicking on the 'Form Maintenance Page' toolbar button  .



The screenshot displays the 'Form Installation Maintenance' page. At the top left is a toolbar icon of a document with a red pencil. The main title 'Form Installation Maintenance' is in green. Below the title is a button with a document and pencil icon labeled 'Modify & reinstall the form'. A section contains the 'Form URL' field with the value 'http://user10.com/forms/Feedbackform/Feedbackform.php' and a dropdown arrow. Below the URL is the text 'Last Installed On: October 15, 2012 11:23 AM' and a globe icon with the text 'Open form page'. A 'More options' dropdown menu is open, showing 'View Form Submissions', 'View log file ...', 'Download data file ...', and 'Download log file ...'. Three red circles with numbers 1, 2, and 3 are placed on the page: circle 1 points to the 'View Form Submissions' option, circle 2 points to the 'Download data file ...' option, and circle 3 points to the 'Open Form Admin Page' button (which has a document and key icon).

The following are the buttons on the form maintenance page.

'Modify & reinstall the form' button

Press this button to switch to the 'form edit' mode where you can add and change the form elements, processing options etc.

Form URL

The list of URLs where you installed this form

'Open form page' button

Open the form online in your default browser.

'Form Admin Page' button

Opens the online form admin page in the browser. You can view & download the form submissions, update database loaded lists and more in the form admin page. The Form admin page is password protected with the admin login your provided in the Take the code step.

'More Options' button

Press the 'More options' button to get the menu that contains more operations.

View form submissions

Download and view the form submission data. Pressing the 'View form submissions'

button opens

The form submissions are displayed in rows and columns. The data view also contains a detail view page as well.

In order to do View form submissions, enable save form submissions to file or save form submissions to database table option in the form processing options page.

View log file

The form processor script writes a log file while processing the requests. This log file will be useful for troubleshooting purposes. You can enable the logging the 'Form processing options' page. Press the 'view log ' button to open the log file viewer. The log viewer will download and will display the log.

Download data file

Opens the Download form submission data file dialog box. Here you can download the form submission data file from the server.

'Download log file' button

Press the 'Download log file' button to open the dialog box where you can download the log file.

See: Download the log file dialog box

Form Admin Page

Form admin page is to administer the form online.

The Form admin page:

- displays the form submissions online
- allows you to download the form submissions in a gzip compressed file
- allows you to edit the dynamically loaded lists (Dropdown lists with load from database option enabled)

 The form admin page displays the form submissions online only if you have enabled 'Save to Database table' option in the 'form processing options' page of Simfatic Forms.

Form admin page login

Only the admins should be able to view form submissions or make updates to the form. So, the form admin page requires login. You have to create the admin login in the Take the code -> 'Admin Page Settings' step.

Opening the form admin page

Go to the form maintenance page by pressing this icon



In the maintenance page, there is a "Form Admin page" button. Press the button to open the Form Admin page in the browser.

Sections in the form admin page

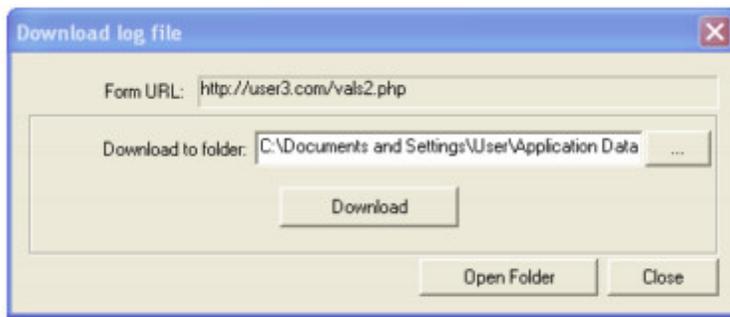
- Form submissions
- Lists

See Also:

How to view form submissions online

Download log file dialog box

In the Form installation maintenance page choose the 'Download log file' option (press the 'More Options' button for menu) to open the 'Download log file' dialog box.



The form processor script writes a log file as it processes the requests. The log file would be useful in troubleshooting situations.

See Also:

- 'Enable logging' option in the Form Processing Options Page
- Turn logging off for better performance

Downloading the log file

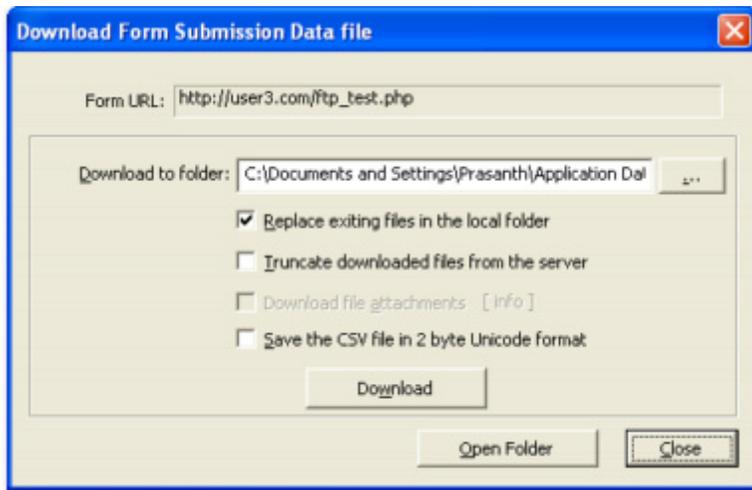
Select the destination folder to download the log file ('Download To folder'). Then press the 'Download' button to start downloading the files.

Open Folder

Press the 'Open Folder' to open the folder where the log file was downloaded.

Download form submission data file dialog box

In the Form installation maintenance page choose the 'Download data file' option (press the 'More Options' button for menu) to open the 'Download form submission data file' dialog box.



You can download the form submission data file (As a CSV - comma separated values- file) in this dialog.

See Also: 'Save to File' option in the Form Processing Options Page

Downloading the data files

Select the destination folder to download the data files ('Download To' folder). Then Press the 'Download' button to start downloading the files.

There could be more than one form submission data files (the data file gets truncated on reaching a size limit).

Replace the existing files in the local folder

If the destination folder contains a file with the same name as the downloaded file, it gets replaced by the newly downloaded file.

Truncate downloaded files from the server

Truncates the data file in the server after download. This way, you can save space in the server and reduce the data file download time when you download it next time.

Download file attachments

This is applicable only if you have one or more file upload boxes in your form.

The files that are attached while submitting the form are all saved in a folder in the server side. If you select the 'Download file attachments' option, all the attached files also are downloaded along with the data file.

If 'Truncate downloaded files from the server' option is selected, then the file attachments are removed from the server after download.

The file attachments are downloaded to a sub-folder under the data file download folder.

Save CSV file in 2 byte Unicode format

Use this option if your form submissions contain non-English characters. Simfatic Forms by default saves the csv file in UTF8 format. Some CSV reader applications can't display CSV file in UTF8 format properly. If this happens, select the 'Save CSV file in 2 byte Unicode format' option.

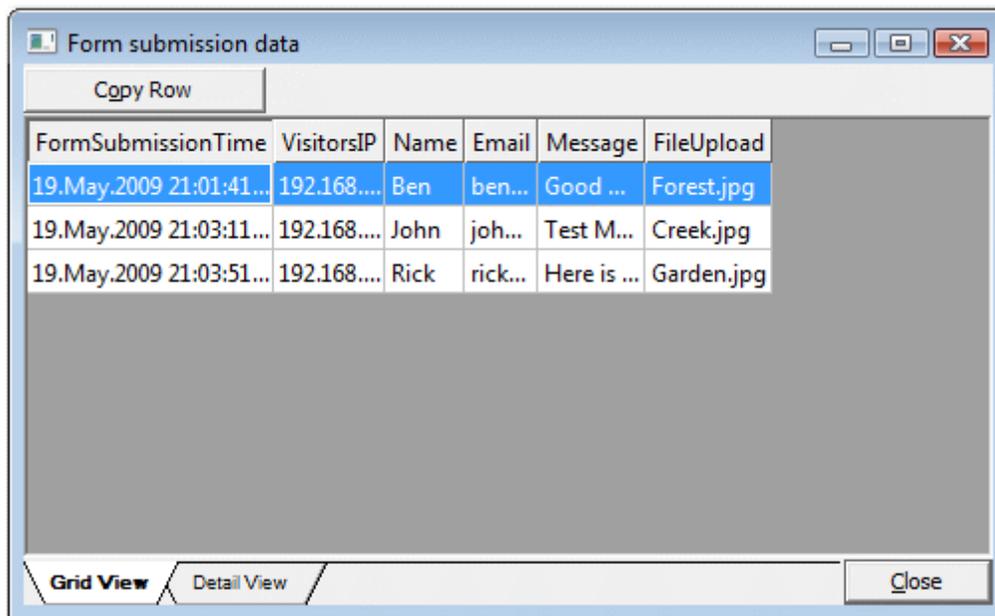
Open Folder

Press the 'Open Folder' to open the folder to which the data files were downloaded.

Form submission data viewer

You can browse through the form submissions data in the data viewer.

In the Form installation maintenance page, press the 'view form submissions' button to open the data viewer.



Grid View

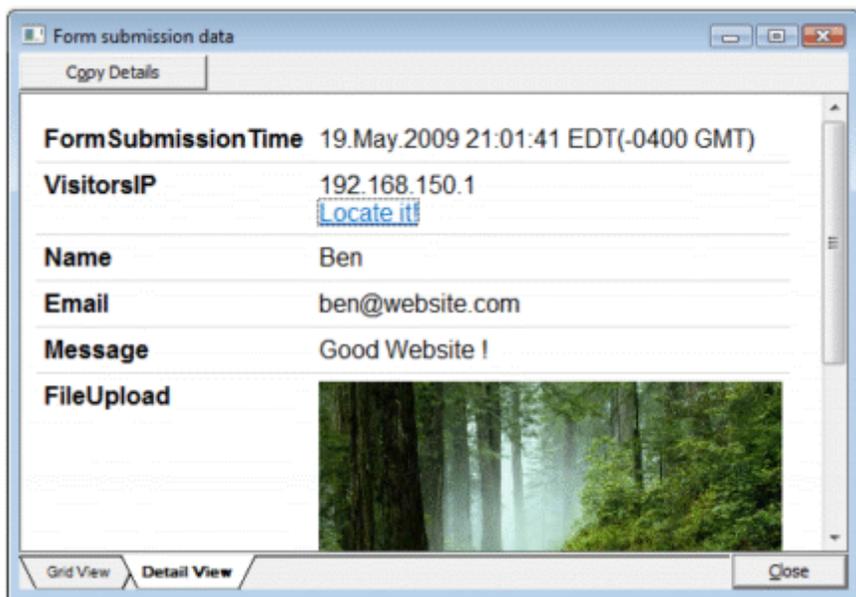
The grid view shows all the form submissions in rows & columns. Each row contains one submission.

Copy row

To copy a row, select a row and then press the 'Copy Row' button.

Detail View

Select one row and press the 'Detail view' tab to see more detailed view of the submission.



Copy Details

To copy the detailed submission, select the detail view and then press the 'Copy Details' button.

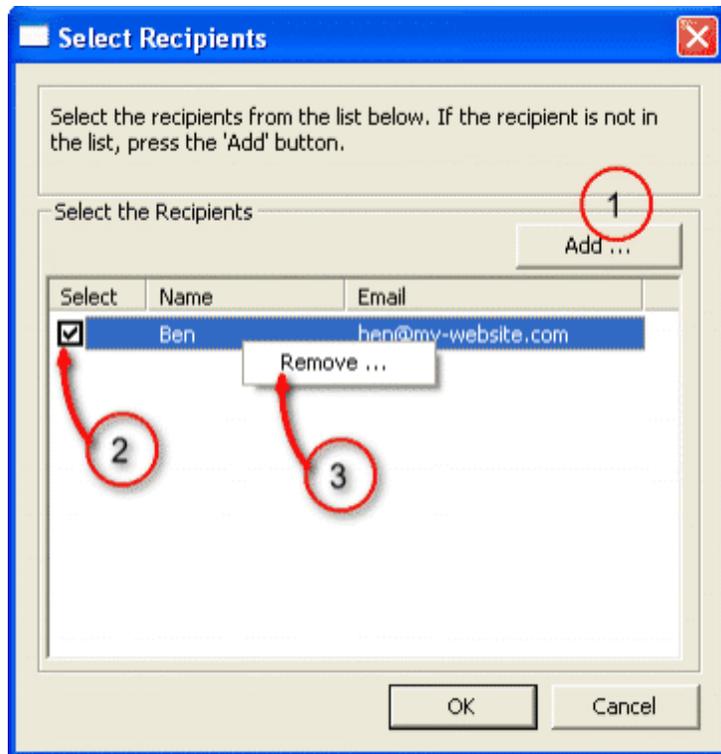
5.20 Common Dialog Boxes

This section contains reference for Dialog boxes and other screens commonly shown in different steps of Simfatic Forms

Address Book

Address book

The address book is to select the email addresses. Press the 'select' button in the 'form to email' page to open the address book.



1. 'Add' button

Press the 'Add' button to add new email addresses to the list. Pressing the 'Add' button opens the 'Add Email' dialog box where you can add the new Name and email address.

2. Select the Recipient addresses

Check the box to select the address to the list of recipients.

3. Removing addresses

To remove one or more addresses, select the addresses and right click. The context menu that opens contains the option to remove the addresses.

Insert Field Table

The field table dialog box allows you to customize the field table and insert in to the template (like the form to email page, confirm page etc)

The field table box uses a simple syntax so that you can easily format the field table

Here is a sample field table :

```
---Page 1
```

```
Name      | %Name%  
Email     | %Email%  
Comments  | %Comments%
```

```
----More Info:
```

```
FormSubmissionTime | %_sfm_form_submission_time_%  
VisitorsIP        | %_sfm_visitor_ip_%  
UniqueID          | %_sfm_unique_id_%
```

Headings

A line starting with two or more hyphens becomes a heading.

For example:

```
--Heading 2
```

```
---Heading 3
```

```
----Heading 4
```

Tables

Tables are created by having the columns separated with |

Example:

```
Name      | %Name%
Email     | %Email%
Comments | %Comments%
```

Auto generated table of non-blank fields

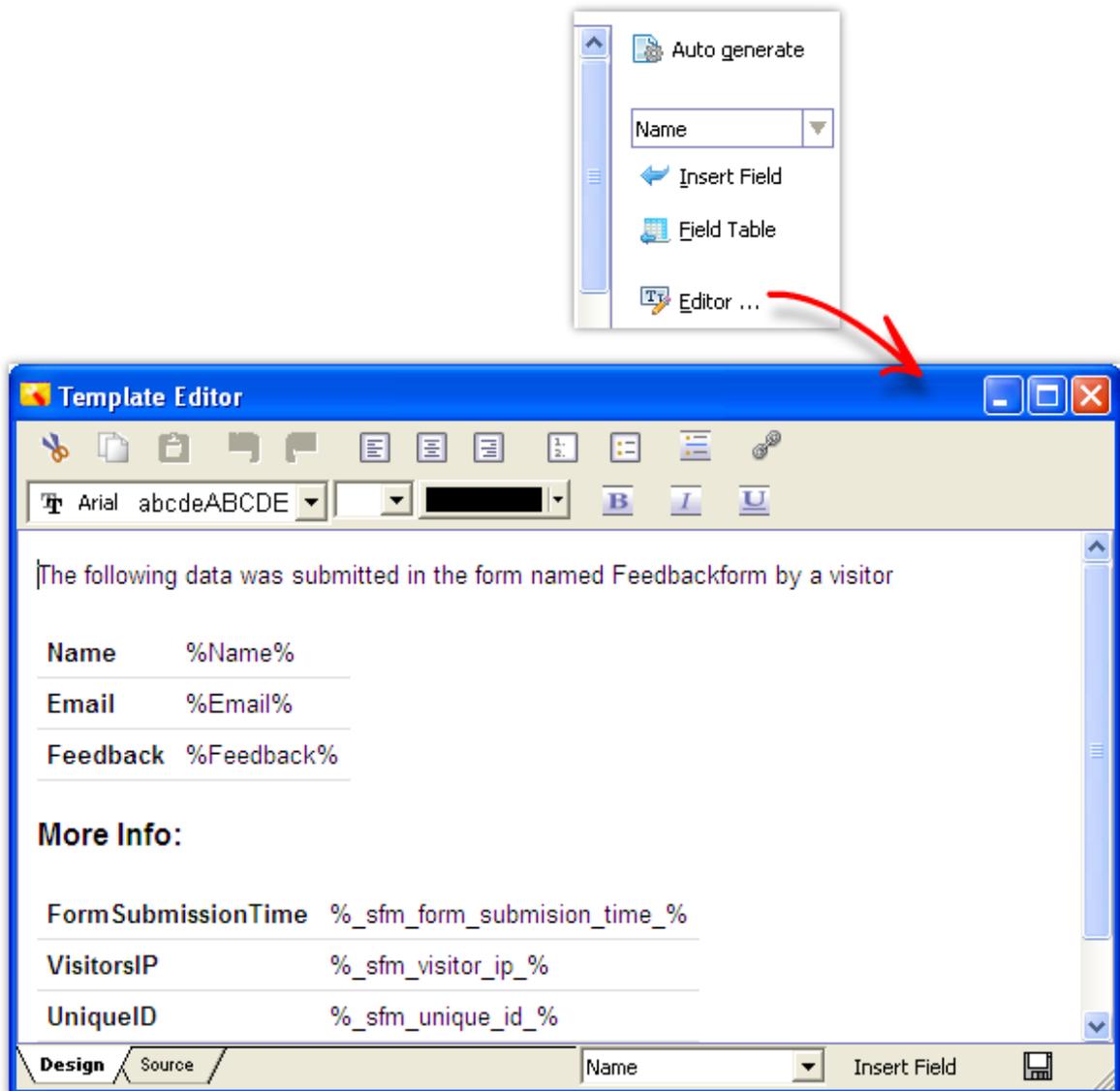
if you choose this option, the field table is generated in run time, putting only the non-blank fields, in a simple table.

This option is useful when there are many optional fields in the form.

HTML Editor

The HTML editor is for editing HTML content like the Form mail template page, the Auto response template, the Thank you page template etc.

Press the 'Editor' button to open the content in the editor window.



1. Design

Select the 'design' tab to edit the content in WYSIWIG (What You See Is What You Get) mode.

2. Source

Select the 'source' tab to edit the HTML source. You can manually edit the HTML tags as required.

3. Insert Field

Insert a form field into the content. Choose the field in the drop down list and then press the 'Insert field' button.

4. Save

Save the changes in the editor.

5. Editor toolbar

Do editing operations like change the font, make the text bold, align etc

Find Element Dialog

Press the Find element button  to search for an element in the form.



Type in the first few letters of the field to search for the element.

The Find element button is available in the 'Draw the form' page (standard tab) and in the input validations page

5.21 Imported Forms

You can import form from an existing web page into Simfatic Forms.

After importing, you can add validations and form processing support to the form.



You can not edit the form elements in the imported form.

Read form from a web page

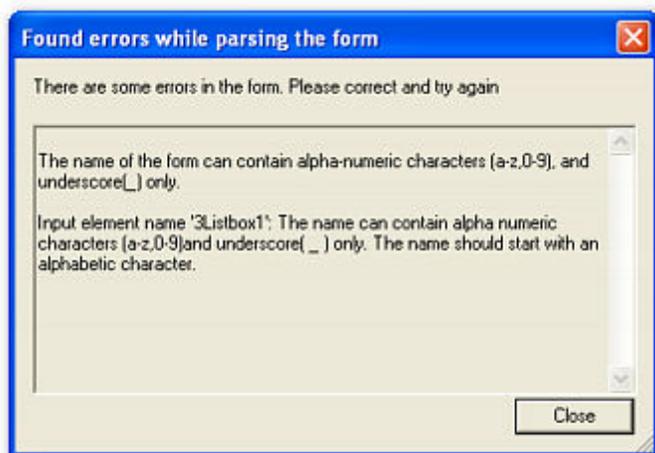
You can let Simfatic forms grab a form from an HTML file. Once the form is parsed out from the page, you can add validations, attach server side processing, install and manage the form using Simfatic Forms.

Read form from a web page

In the The First Page, press 'Add validations and form processing support to an existing form' or 'Add validations to an existing form' button and select the HTML file.

If there are errors while parsing the form, Simfatic Forms will display the errors.

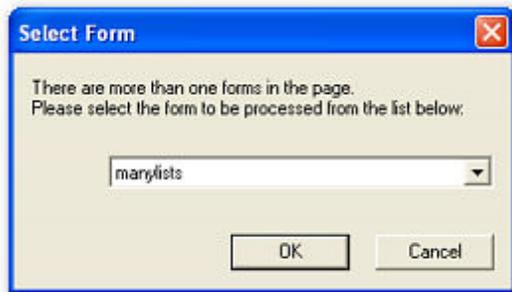
Simfatic Forms ensures that the form follows the proper HTML rules. Although the form page displays fine in the browser, it is required to follow these rules for the proper processing of the form. For example, the form should have a name and the name of the form and the input elements should follow Certain Naming Rules



If there are no errors, you will be taken to the 'Select general options' page.

Pages with more than one forms

If more than one forms in the same page, SimfaticForms will prompt you to select a form.



Once a form is successfully parsed out from the file, you will be taken to the General Options Page

General Options

This page is displayed only if you pulled out the form from an HTML file using the import buttons in the The First Page.

Form Name(Read Only)

The name of the form that is read from the web page. The form name is used to identify the form through out the cycle (in the form page, server side processing, in the response mails, form submission data file and so on). It is recommended that you give a unique name for each of your form.

Server side processing support

If you want the Simfatic form processor to process the submitted form data, select the first option.

If you just want to add client-side input validation, select the second option. In this case, you will have to use your own server side script to process the submitted form data.

Look & Feel Support

You can decide whether you want to customize the error string display (The error strings that are displayed if the form validations fail).

1. Standard

Select this option if you don't want to customize the style and let Simfatic Forms use the standard style.

2. Custom

Select this option if you want to customize the style. If you select this option, the 'Look & Feel' page will allow you to edit the visual style.

Index

A

- add server wizard 319
- addressbook 339
- advanced code generation options 317
- align 214
- alphabetic validation 271
- alphanumeric validation 271
- anchor 252
- auto-generate 341
- auto-response 51, 296

B

- background color 111, 211
- blank form 20, 193
- blog
 - integrate form 116
- border 111, 211
- box 256
- button
 - reset 240
 - submit 237

C

- calculation field 139, 242
- calendar 247
- captcha 249
- change form 53
- checkbox 221
 - group 223
- code generation options 317
- color 111, 211
- combo box 225
- comparison validation 271
- conditional
 - display elements 54
 - validation 84
- confirmation page 294
- copy 214
- copy code 315
- copy form code 315
- create form using template 196
- create new 20
- create new form 193
- csv downloading 333

- custom 99
 - css 260
 - extensions 99
 - HTML 258, 260
 - javascript 107, 260
 - style 111
 - validations 86
- custom code 260
- custom widget 258
- cut 214

D

- data downloading 333
- database login 183, 311
- database page 301
- date format 207
- date picker 247
- decimal number 221
- default code output folder 202
- deplay form 309
- display elements conditionally 54
- DOCTYPE 202
- download file uploads 329
- downloading csv file 333
- downloading form submission data 333
- Dreamweaver
 - embedding form 118
- dropdown list 225
 - validations 276
- drupal 121
- duplicate 214

E

- edit form 53
- editor
 - HTML 343
- email 289
 - auto-response 51, 296
- email settings 311
- email validation 271
- error message 101
- error string 101
- Expression Web
 - embedding form 119
- extension modules page 306

F

- field table 341
- file 234
- file upload 234
 - native 236
- find elements 345
- font 111
- footer 260
- form admin page 329
- form mail page 289
- form page import 346
- form processing options 285
- form properties 207
- form submission viewer 336
- form template 196
- form to email 289
- form to file page 299
- form working 14
- formatted text 253
- formdata folder 186
- formula 244
- ftp login 319
- full form code 188
- functions
 - calculation 143
- fwz file 133

G

- general form properties 207
- group box 256

H

- header 260
- heading 257
- hidden 236
- html editor
 - email template 31
- hyper link 252

I

- image 252
- imported form 346
- insert image 252

- integrate 116
 - dreamweaver 118
 - drupal 121
 - expression web 119
 - form 116
 - joomla 120
- integration 115
 - website 315

J

- Javascript event handling 107
- joomla 120
 - integrate form 120

L

- label 219
- link 252
- list 225
 - validations 276
- list format 227
- loading list from database 225
- localization 207
- logfile download 331
- logging 184
- look and feel 111

M

- menu 198
- modify form 53
- modify server 323
- multi-line text 233
- multiple choice 231
- multiselect list 224
- multi-select list 224

N

- naming rules 135
- number 221
- numeric validation 271

O

- options dialog box 202

Index

P

page settings 265
parsed form 346
passing a value 102
paste 214
payments 105
paypal 105
permissions formdata folder 186
preview 201
pull down menu 225

R

radio button 231
read from url 347
reference
 calculation functions 143
reply-to 293
reset button 240
response email 51
response page 307

S

save to database 301
scrollable text 114
search elements 345
Simfatic Forms 12
Simfatic Forms project file 133
single line textbox 220
size validation 271
smart validation 156
SMTP 314
subdomain 97
submit button 237
system requirements 12

T

T&C box 114
tab order 263
take the code wizard 311
template editor
 email template 31
terms & conditions box 114
text block 114, 253

textarea 233
textbox 220
thank you page 307
theme 111
tutorial 19

U

unique validation 157
upload 234
 native 236
upload form 309
URL 102

V

validation 266
 based on a condition 84
 condition 283
 conditional 84
validation message 101
validations
 alphabetic 271
 alphanumeric 271
 calculation field 282
 checkbox 274
 checkbox group 275
 comparison 271
 date picker 279
 dropdown list 276
 email 271
 file upload 277
 multi-line text 276
 multi-select list 282
 native file upload 282
 number field 280
 numeric 271
 radio group 275
 size 271
 textarea 276
 textbox 271
view form submissions 329

W

webserver 179
website
 integration 115

X

XHTML 202