Calc Builder

Quick Guide V3.x

Moonsoft

www.moonsoft.es

In this quick step guide we are going to follow basic steps to create a full-operating calculator as Simple Sum. We are going to write a form with 2 number fields and return the sum of these numbers. The manual process of building the calculator is valid for default CalcBuilder extension, please check 'how to build your calculator from Excel' for Calcbuilder Extended.

First of all we install extension package, which automatically installs component and module, and upgrades from previous version if you already had Calc Builder installed.

Troubleshooting:

Failed to load this file in server

This new version of the extension contains a heavy library, regarding pdf creation. Because of it's size, some servers reach the file size limit to upload the zip file, first step of installation. If this is your case, you can try an alternate installation method:

Method 1:

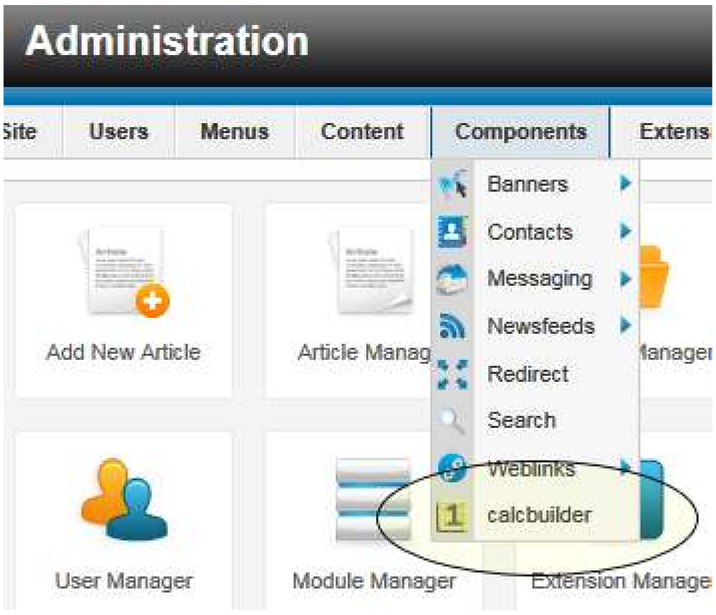
- Upload the zip file manually to any location of your server.
- Select 'Install from URL' option from Extension manager and write the url pointing to the zip file.

Method 2:

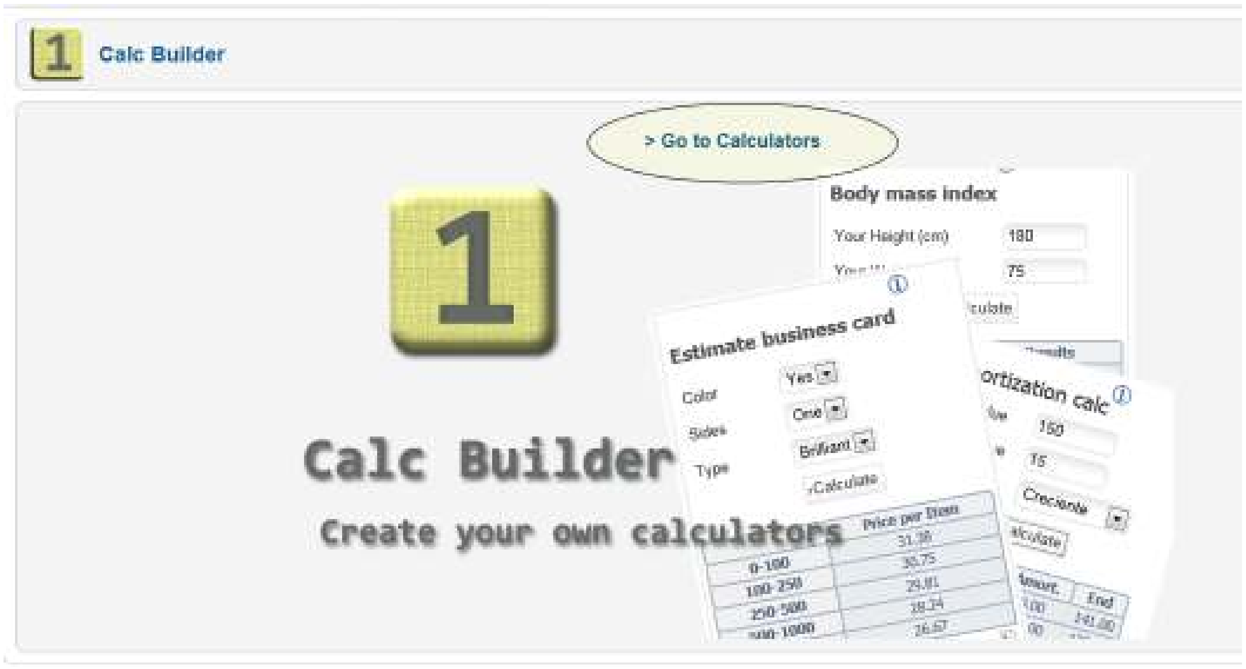
- Unzip and upload contents to a temp folder on your server (by default joomla/tmp).
- Select 'Install from Directory' option from Extension Manager and select the folder where you previously uploaded the files.

1. Calc Builder Menu

Calc Builder has only one menu link

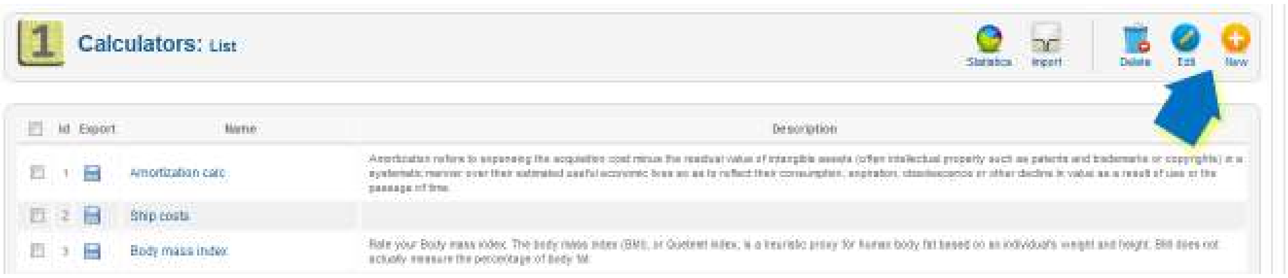


Click in menu link



Click on “>Go to Calculators” or at the image below to go to list of calculators screen.

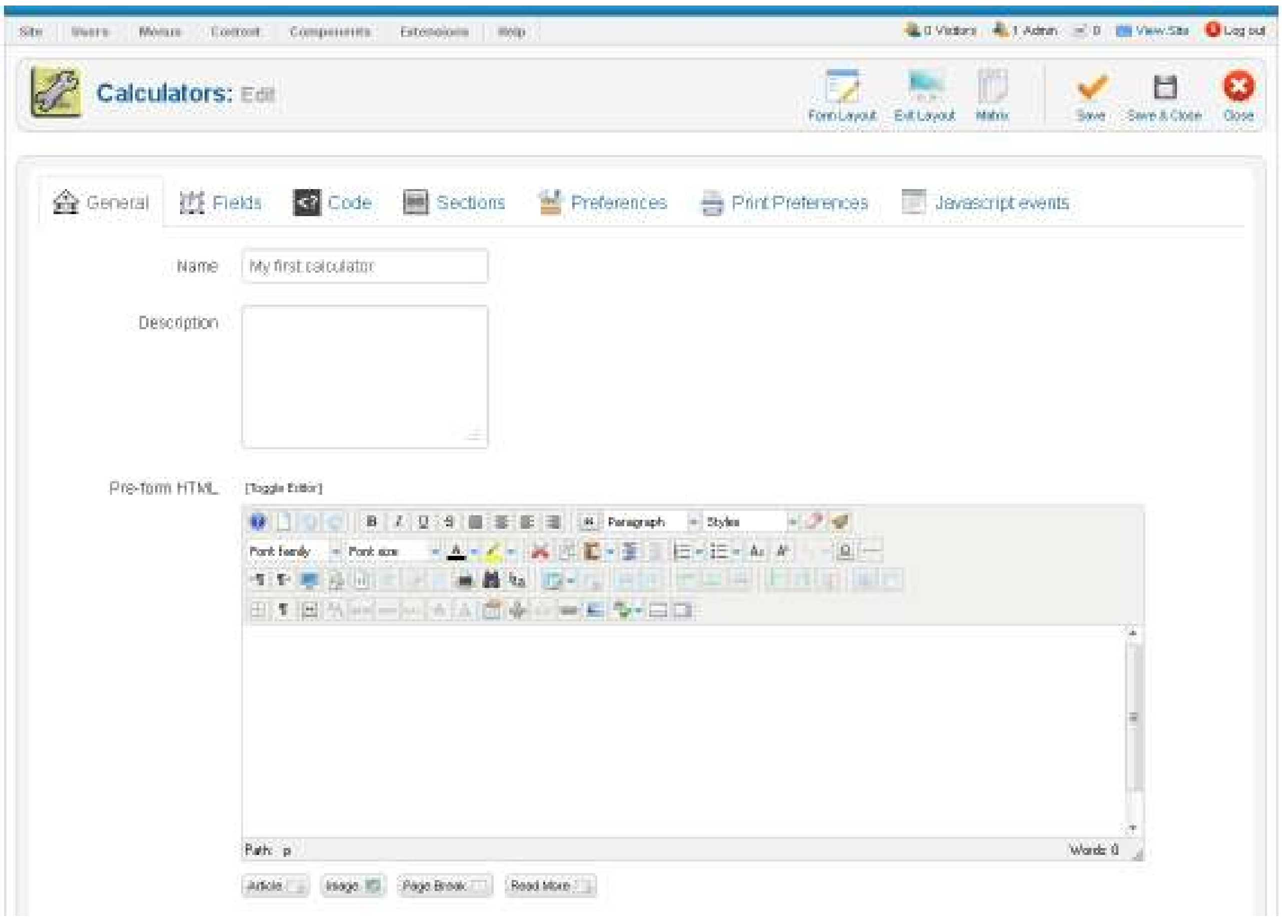
Calculator List



	Id	Export	Name	Description
	1		Amortization calc	Amortization refers to expensing the acquisition cost minus the residual value of intangible assets (often intellectual property such as patents and trademarks or copyrights) in a systematic manner over their estimated useful economic lives so as to reflect their consumption, expiration, obsolescence or other decline in value as a result of use or the passage of time.
	2		Ship costs	
	3		Body mass index	Rate your Body mass index. The body mass index (BMI), or Quetelet index, is a heuristic proxy for human body fat based on an individual's weight and height. BMI does not actually measure the percentage of body fat.

First of all we are going to create a new Calculator.

5. Calculator Edit/New



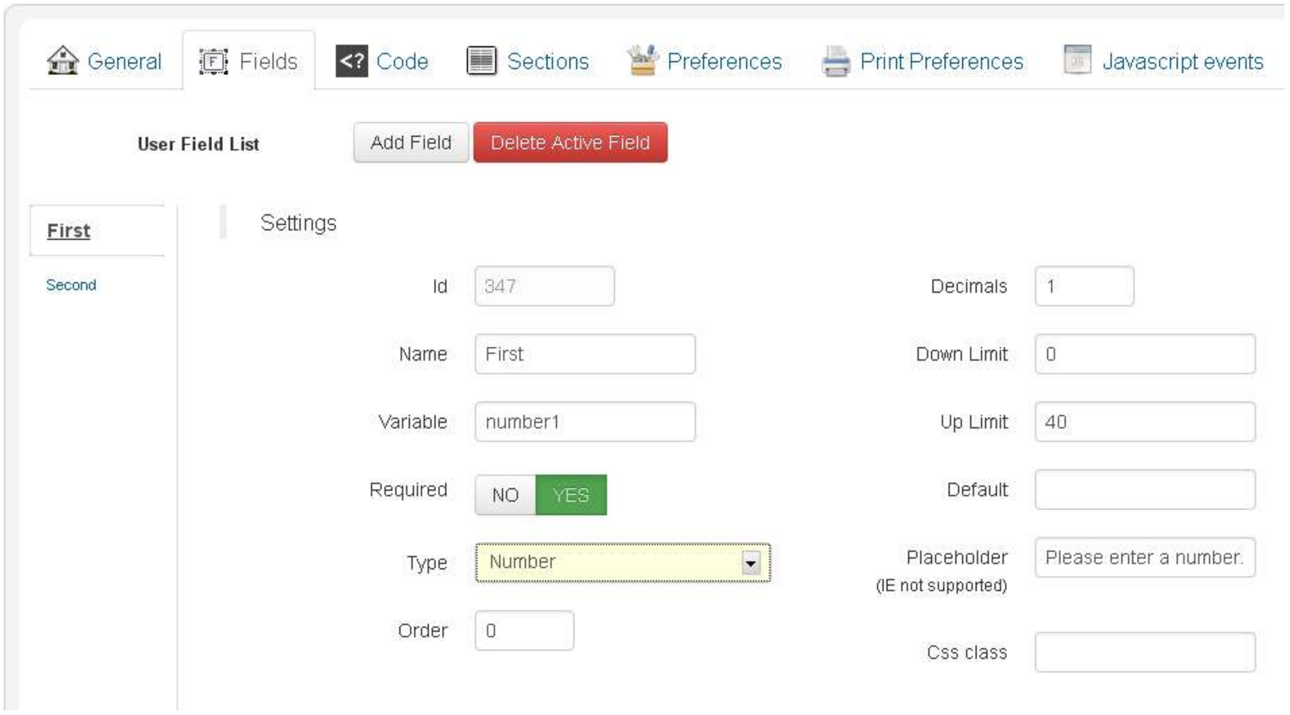
Name: My first calculator

Description:

Pre-form HTML:

Let´s fill in General tab **Name** and **Pre-form HTML**

Now, we are going to declare our form fields switching to Fields tab.



Id	347	Decimals	1
Name	First	Down Limit	0
Variable	number1	Up Limit	40
Required	NO [✓ YES]	Default	
Type	Number	Placeholder (IE not supported)	Please enter a number.
Order	0	Css class	

Clicking in Add field button we add fields to our form. In our example we create 2 number fields. You can set more parameters like number of decimal places allowed, or up/down limits for the number introduced, required fields...all these settings will be transformed to form validations when users fill in the form and press 'Calculate'.

At the next tab, 'Code' we must configure the operations needed following PHP sintax. We'll ask the calculator here to perform a sum of the fields. This code area will be executed everytime frontend users click 'Calculate' button. Input variables, and any other variable created at this time can be used when we write Exit Layout. By now, we write our sum in code:

```
$sum=$num1+$num2;
```

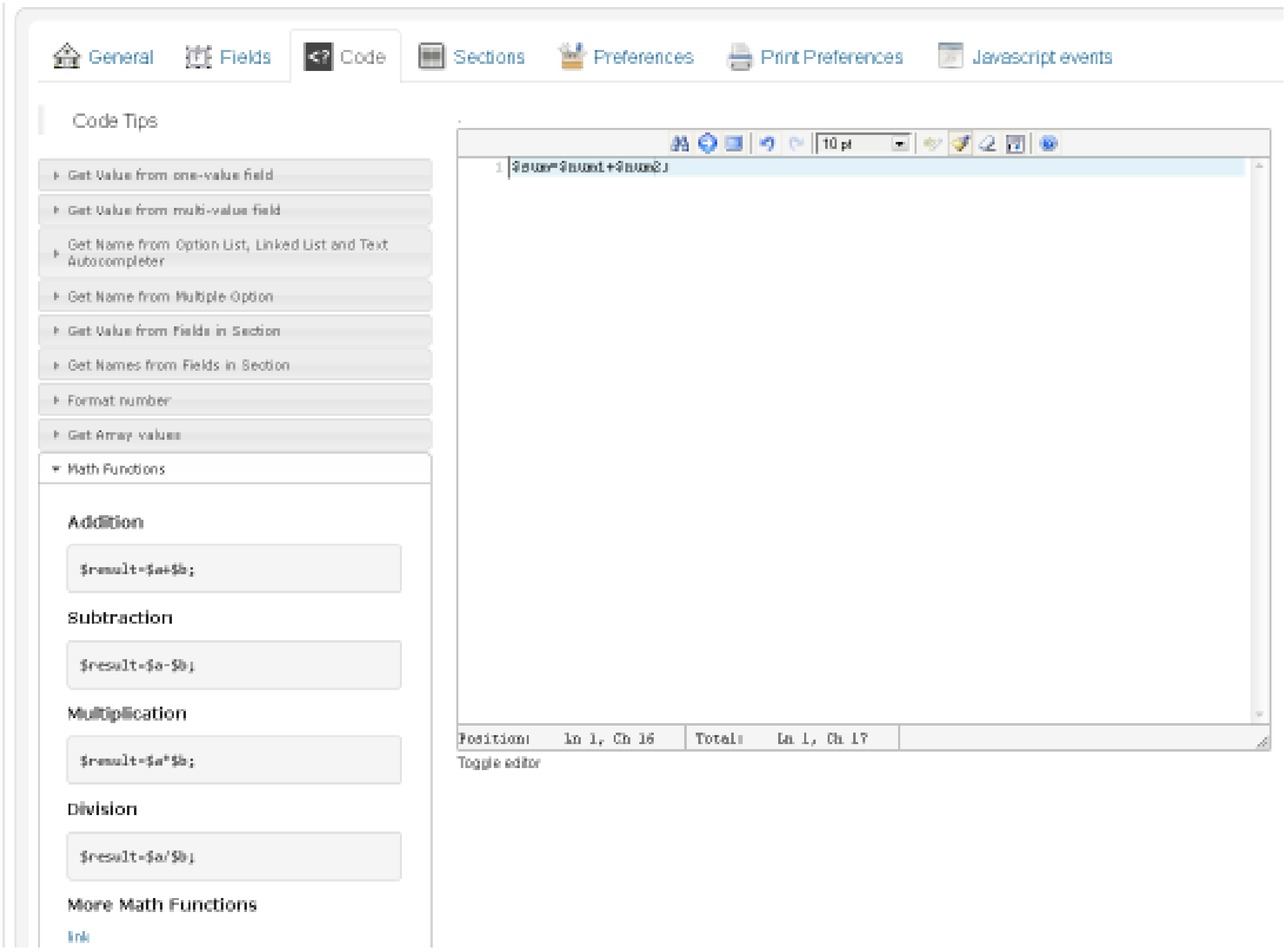
Note that $num1 and $num2 are the variables of our fields. All we´re going to do is sum these numbers. We have stored the result at the $sum variable, which we will print at our output afterwards. In order to configure your operations, you can write any valid math operation, just preceding variables with **'$'** sign, and finishing each calculation line with a semicolon ';'. You can check available simple math operators here:

http://www.w3schools.com/php/php_operators.asp

Also you have available any valid function following php sintax, so you can use any other function or expression, like conditional blocks:

http://www.w3schools.com/php/php_if_else.asp

or any other valid php code. At the left panels you will find information to help you to recover and handle property all types of input fields from your form.


General
Fields
Code
Sections
Preferences
Print Preferences
Javascript events
Code Tips
Get Value from one-value field
Get Value from multi-value field
Get Name from Option List, Linked List and Text Autocompleter
Get Name from Multiple Option
Get Value from Fields in Section
Get Names from Fields in Section
Format number
Get Array values
Math Functions
10 pt
Toggle editor
link

```
$sum=$num1+$num2;
```

Position: ln 1, Ch 16 Total: ln 1, Ch 17

Addition

```
$result=$a+$b;
```

Subtraction

```
$result=$a-$b;
```

Multiplication

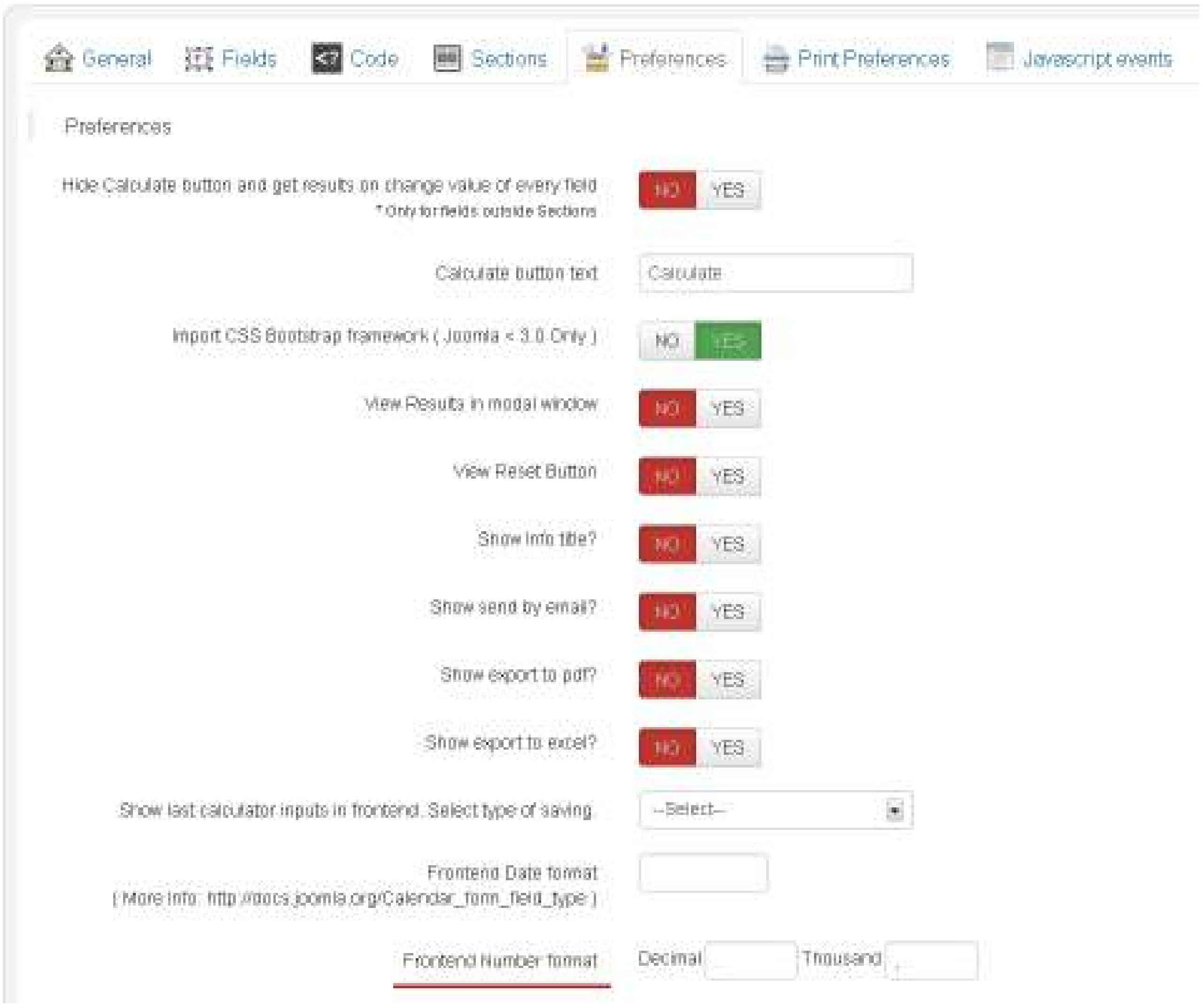
```
$result=$a*$b;
```

Division

```
$result=$a/$b;
```

More Math Functions

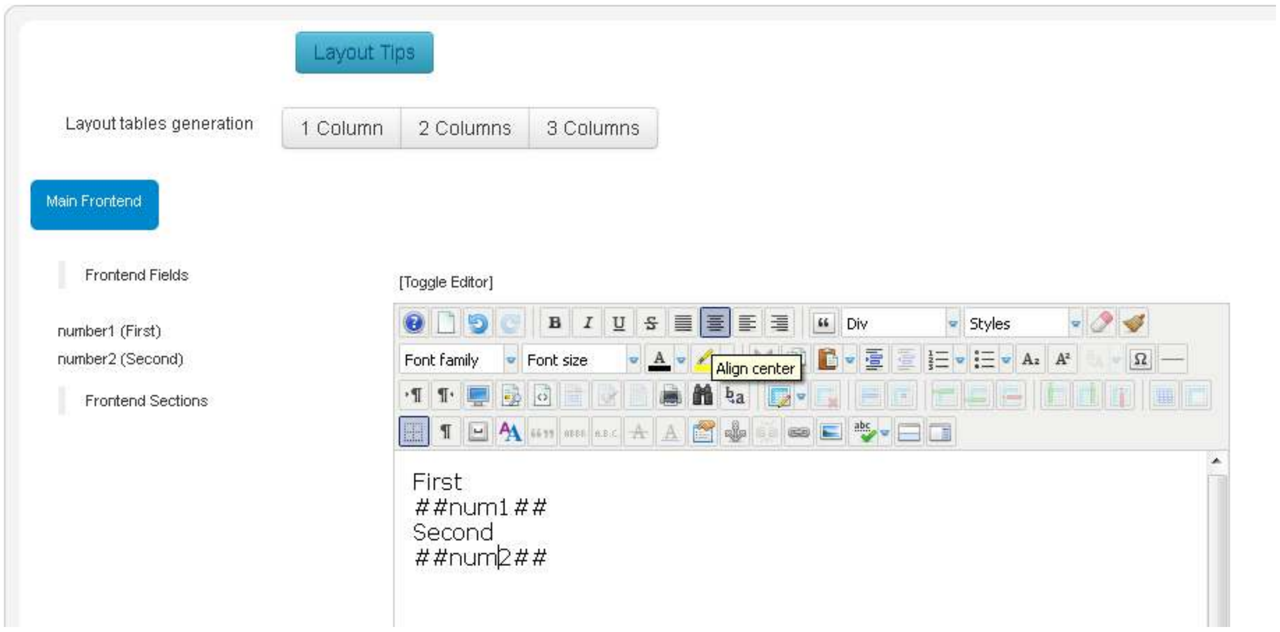
When we use Number fields we must set **Frontend Number format** at preferences tab. You can use . or , for decimal or thousand separator.



Preferences

Hide Calculate button and get results on change value of every field * Only for fields outside Sections	[✓ NO] YES
Calculate button text	Calculate
Import CSS Bootstrap framework (Joomla < 3.0 Only)	NO [✓ YES]
View Results in modal window	[✓ NO] YES
View Reset Button	[✓ NO] YES
Show Info title?	[✓ NO] YES
Show send by email?	[✓ NO] YES
Show export to pdf?	[✓ NO] YES
Show export to excel?	[✓ NO] YES
Show last calculator inputs in frontend. Select type of saving	--Select--
Frontend Date format (More info: http://docs.joomla.org/Calendar_form_field_type)	
Frontend Number format	Decimal Thousand

We´ve got our fields and now it´s time to define our Form layout. But, first let´s save all we´ve done by click in **save** button in toolbar. Then go to **Form Layout.**

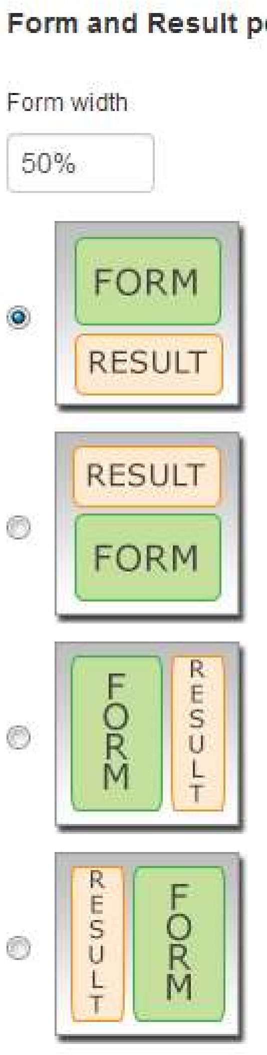


```
First
##num1##
Second
##num2##
```

At the left side you can see fields added at 'Fields and Code'. You can write any HTML layout you need. One good solution is to define a table. To add fields to HTML just write var name between ## . In our example we have **num1** and **num2** , so we write **##num1##** and **##num2##** where we want to see input fields in frontend. You can also create an automatic layout of 1/2/3 columns using the buttons at the top section. These options will arrange your input fields in order using the number of columns specified.

***Note**: the layout editor will use your standard joomla html editor selected*

You can add to your layout as many content as needed, like static texts, images, and every other content you are able to create using your html editor.

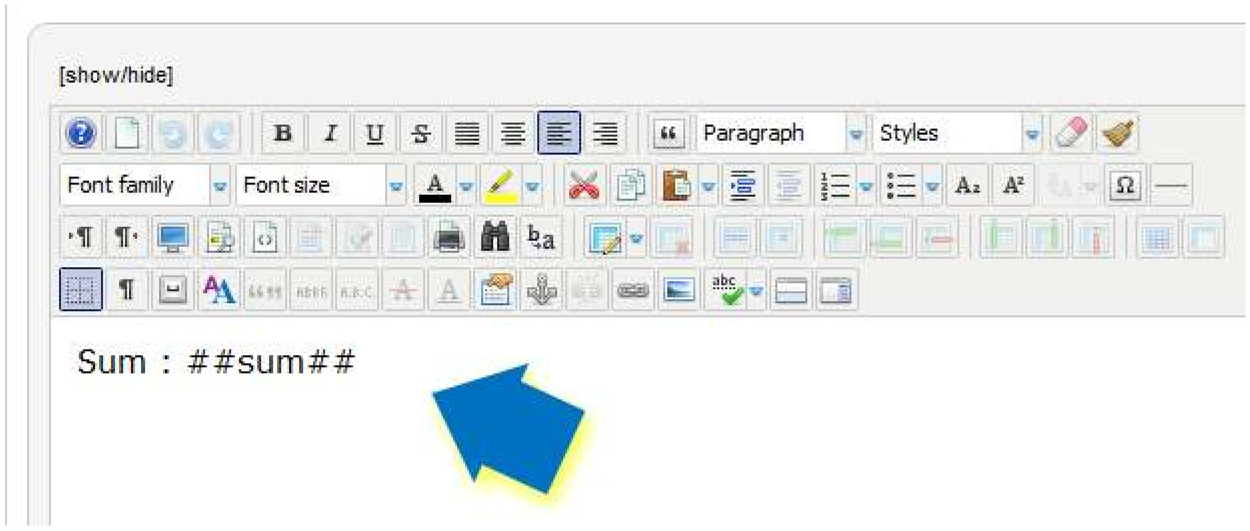

Form and Result position

Form width

50%

At the right side we could select form and result position. You could choose also the width of form part to adjust to your site.

Ok!! Now we save&close and go Exit Layout. At Exit Layout we are going to define our exit template.



Sum : ##sum##

To print variables created at code section, use ##variablename##. In code we´ve written:

```
$sum=$num1+$num2;
```

So, we want to show in exit layout $sum value. Just write ##sum## in HTML exit layout and we will see the result at the Frontend.

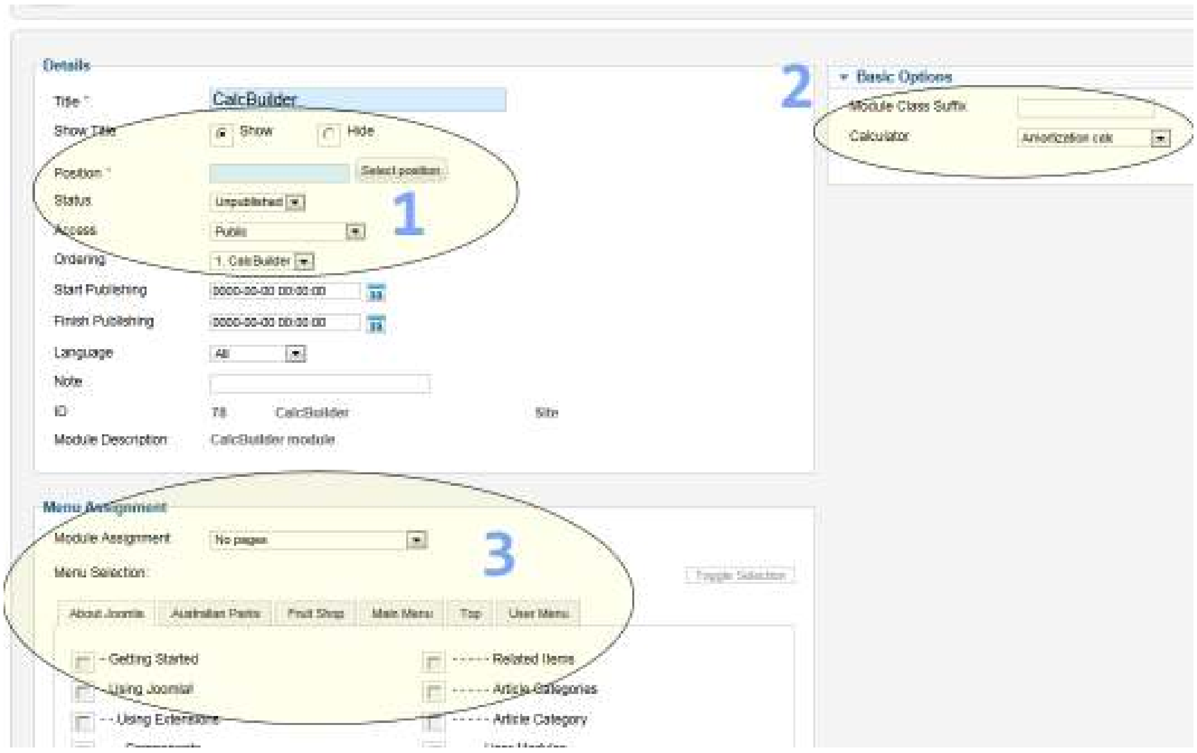
We´ve done our calculator. Now we´re ready to show it at our page.

6. Insert Calculator in our pages

We can insert our calculators in two ways: using a default module standard position or insert one or more modules inside an article.
You can use more than one calculator in one page. Calculators are code independent.

A. Module

In module manager in joomla we can see CalcBuilder. You can create as many calculator modules as you need.
(This screen may differ from Joomla versions, but startup is the same)



Title *	CalcBuilder
Show Title	[✓ Show] Hide
Position *	
Status	Unpublished
Access	Public
Ordering	1. CalcBuilder
Start Publishing	0000-00-00 00:00:00
Finish Publishing	0000-00-00 00:00:00
Language	All
Note	
ID	78 CalcBuilder Site
Module Description	CalcBuilder module
Module Class Suffix	
Calculator	Amortization calc
Module Assignment	No pages
Menu Selection:	

- Getting Started
- Using Joomla!
- Using Extensions
------ Related Items
------ Article Categories
------ Article Category

First step is to fill Module requirements (1): Title, show title, status (must be published), position and other features. In order to get the module shown inside an article with {loadposition XXX} command, you can write your own custom position instead of selecting one available at your template.

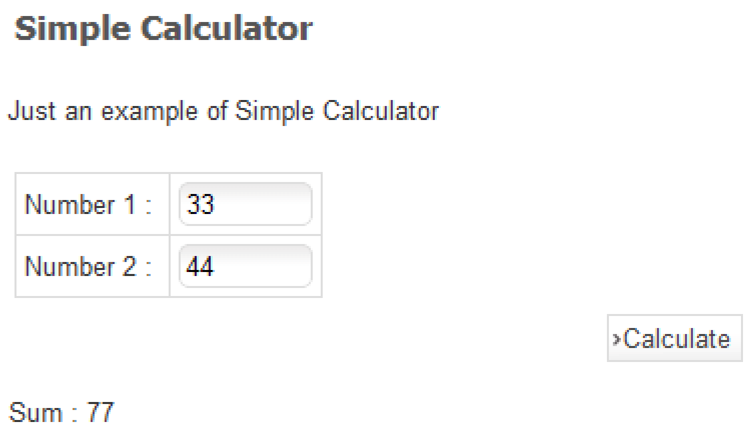
At Basic Options section (2) we select which calculator we want to show in this module.

And finally we select in which pages we can see our module.

B.Article

Inside an article you should use {loadposition XXX} where XXX is the custom position(1) we have defined for the module.

7. Results


Simple Calculator

Just an example of Simple Calculator

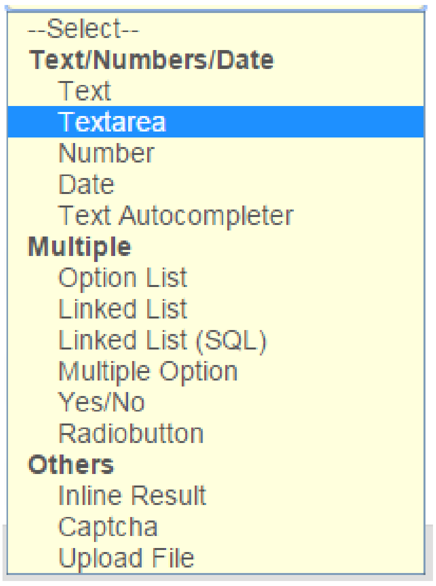
Number 1 :	33
Number 2 :	44

Sum : 77

***Note**: This is the end of quick guide. The rest of this document is intended to go on the detail of all features available for the extension, along with more complex tasks and configuration. Now you can choose to play with the examples and try to build your own calculator or continue reading to get a deeper view of all options available.*

Field types

For our first example we have used the most common field type to be handled in a calculator, the numeric type. But you can include several other types at your input form as well. These are all field types available:

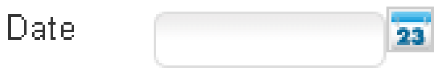

--Select--
Text/Numbers/Date
- Text
- Textarea
- Number
- Date
- Text Autocompleter

Multiple
- Option List
- Linked List
- Linked List (SQL)
- Multiple Option
- Yes/No
- Radiobutton

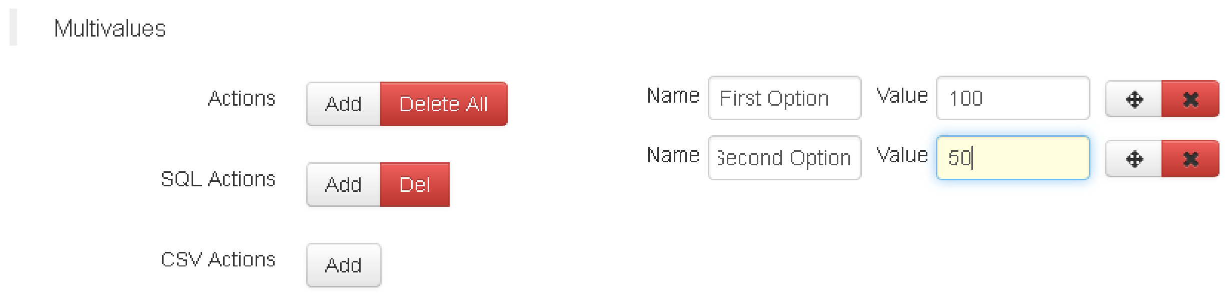
Others
- Inline Result
- Captcha
- Upload File

Number: Shows a default input text restricted only to numbers, performing validations about decimal places allowed, and maximum/minimum values allowed.

Date: Shows a date input followed by default joomla calendar chooser. It will validate date format against the format set at Preferences. (See preferences configuration below)



Option list: Combo box with a single option to choose. You can manually set the options available adding values to the multivalues section, use a sql query to recover dynamic values from your database, or load a csv text to create the options.


Multivalues

Actions Add Delete All

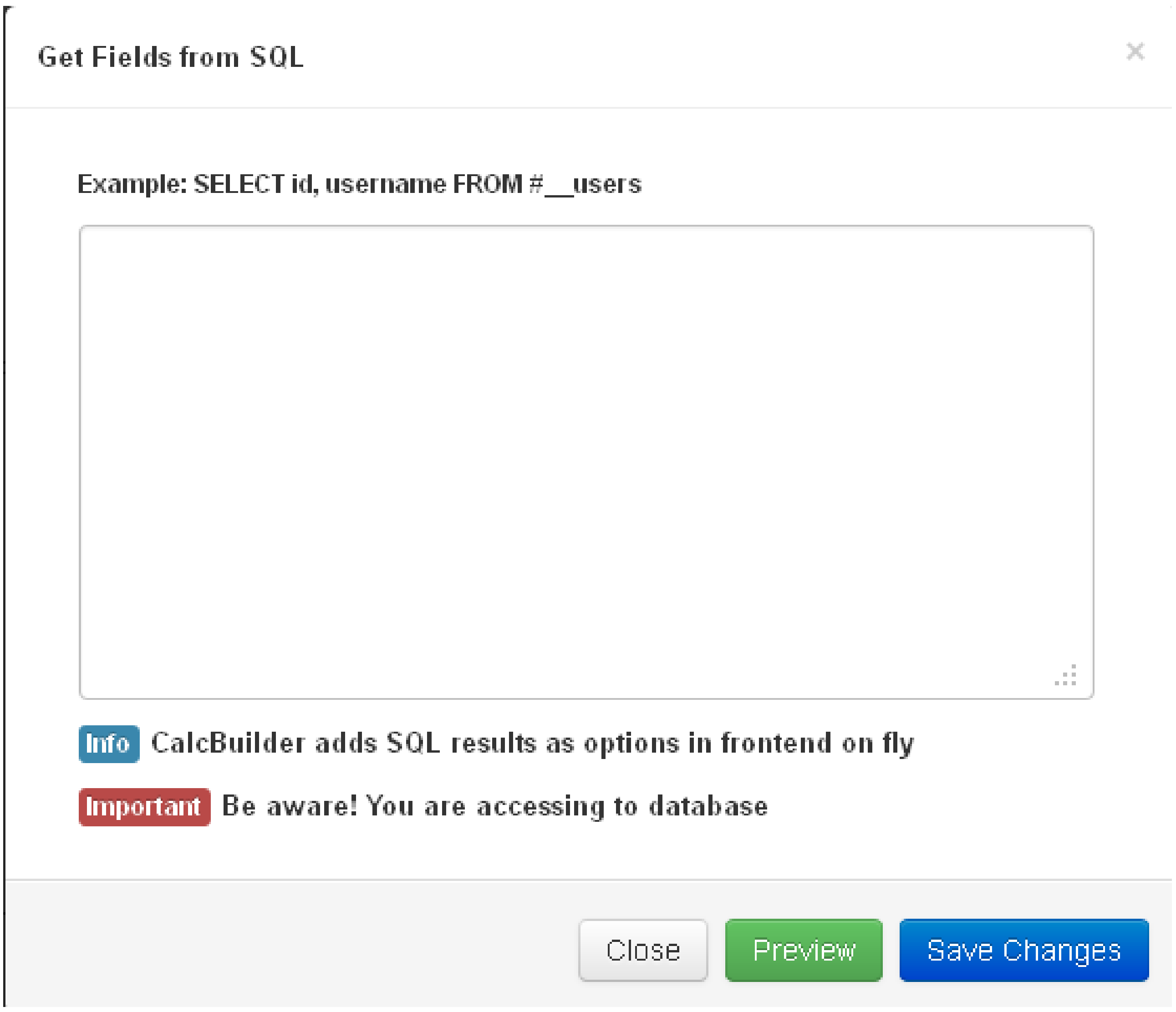
SQL Actions Add Del

CSV Actions Add

Name First Option Value 100

Name Second Option Value 50

or


Get Fields from SQL

Example: SELECT id, username FROM #__users

Info CalcBuilder adds SQL results as options in frontend on fly

Important Be aware! You are accessing to database

Tips/Tricks

When using SQL query to recover values, you can use #_ to replace your database tables preffix, so your query will still be valid when exporting calculators from one site to another or changing the preffix.

For each option you must enter two values, it's name (the option label user will see at the dropdown), and the value of the option (the value you want to recover to use at calculations).

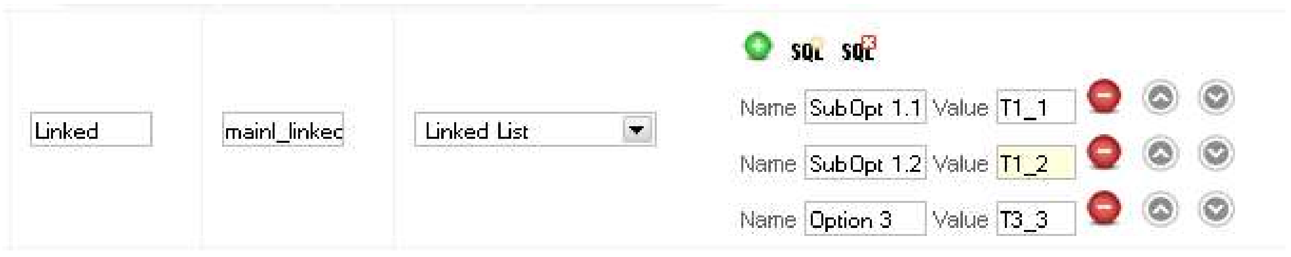
When option 1 is selected, the variable at code section
$mainl will have a value equal to "T1"

You can also recover the label selected at the combobox, another variable named

$xxx_name   (where xxx is the variable name of the list)
is available with the label selected, for this example at the code we can get

$mainl_name to get the label "Option 1"

Linked list: This is an advanced input type. It is an special kind of dropdown list, that filters its options depending on a previous option list selection. To configure the linked list you must follow a naming convention, as follows:

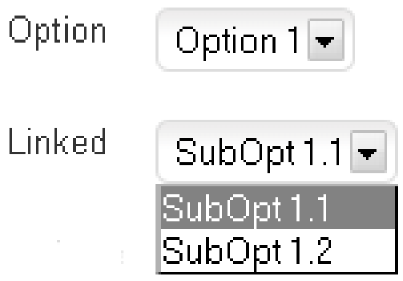


Linked	mainl_linkec	Linked List	Name	SubOpt 1.1	Value	T1_1
			Name	SubOpt 1.2	Value	T1_2
			Name	Option 3	Value	T3_3

You must name its variable like xxx_linked, where xxx is the name of the main list we want this linked list to be dependant on. We had already a default option list named 'mainl', so this one will be named as mainl_linked.

Its options will be filtered according with the value selected from the first list. Each option must have a value named as zzz_yyy where zzz is the value of the first list which enables this option, and yyy is the unique value you want to set for the option.

With this configuration, 'SubOpt 1,1' and 'SubOpt1,2' options will only be available when 'Option 1' is selected from the first list. 'Option 3' will only be available when 'Option 3' of the first list is selected.



Following same structure than default option lists, you can recover the label selected at this linked list using $xxx_linked_name ($mainl_linked_name for this example).

Tips/Tricks

When using linked lists, the value recovered is the one set at 'value' of each option, so you will get the whole string “zzz_yyy” (“T1_1”) when using $mainl_linked variable. In order to split the value from the main list option and get only the unique text for this option, you can use:

$myvalue=explode(“_”,$mainl_linked)[1];

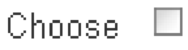
This expression will place the value “1” at your new variable “myvalue”;

Linked list(SQL): This special linked list is able to query the database to recover valid options when according to other values of the form. For this type you will have to select the 'trigger field', which is other field of your form that launches the update of the linked list when changed. And also write down the query to the database to recover valid options, where you can use other values of the form using the ## sintax, for ex:

```
select id,name from #__users where name = ##fieldname##
```

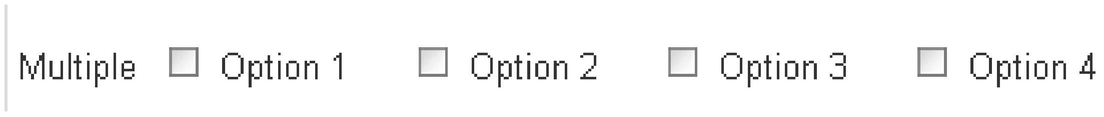
This query will fill your option list with all users named after 'fieldname', that can be any other input field of your form.

Yes/No: It will show a single check to select.



When recovering value of this field type you will get “Y” if the check was selected, no value otherwise.

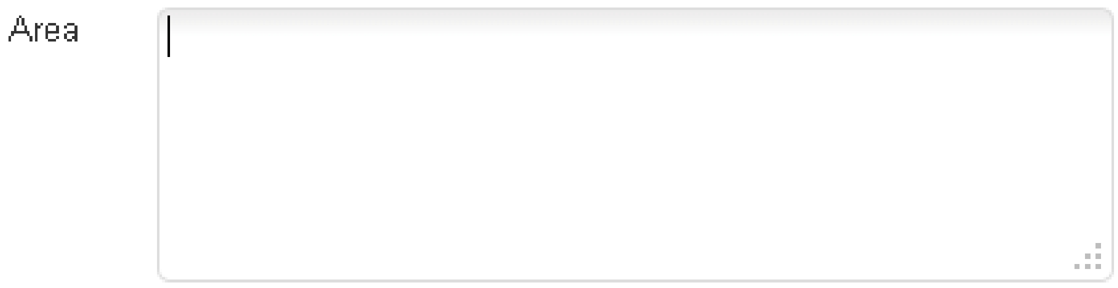
Multiple option: This field type will create as many checkboxes as options included. You must configure the options as you do with a option list, with a label and a value, and you will get this output.



To recover options selected, you will receive a list of options checked, in form of array type. So you should access the array using a loop or counter, if we named our multiple field as 'multiple', then we will have at $multiple[0] the first value selected, at $multiple[1] the second option selected, and so on. We can find out how many options are selected using php function count
count($multiple)
to have the number of checkboxes checked.

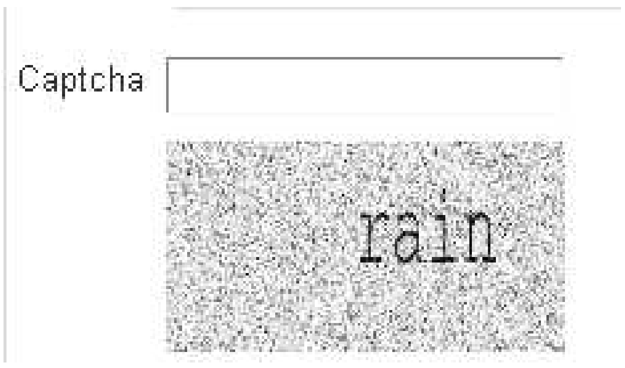
Text: A single line text

TextArea: A multiple line text.

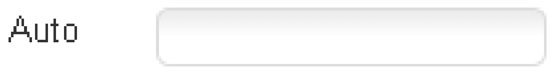


Inline result: this is a result that you want to show inside the input form. Just place it at the form layout as any other input, and assing it any value at the code section. You will see each time the input is changed, the inline results will recalculate themselves and show the result automatically at the input form without pressing calculate button.

Captcha: Used to ensure human beings are using your form, you can add this field to show a text validator it will be executed automatically when sending the form.

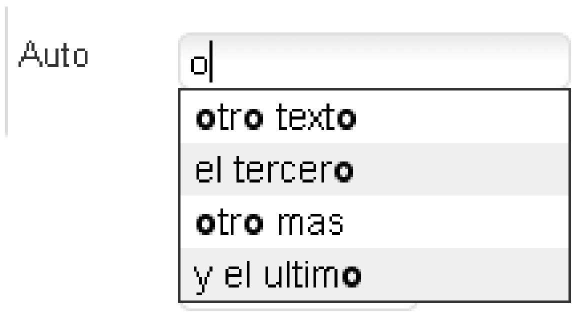


Text Autocompleter: This is another advanced subtype of option list. When your option list contains a high number of entries, you can help your users to filter them using this type. You must configure

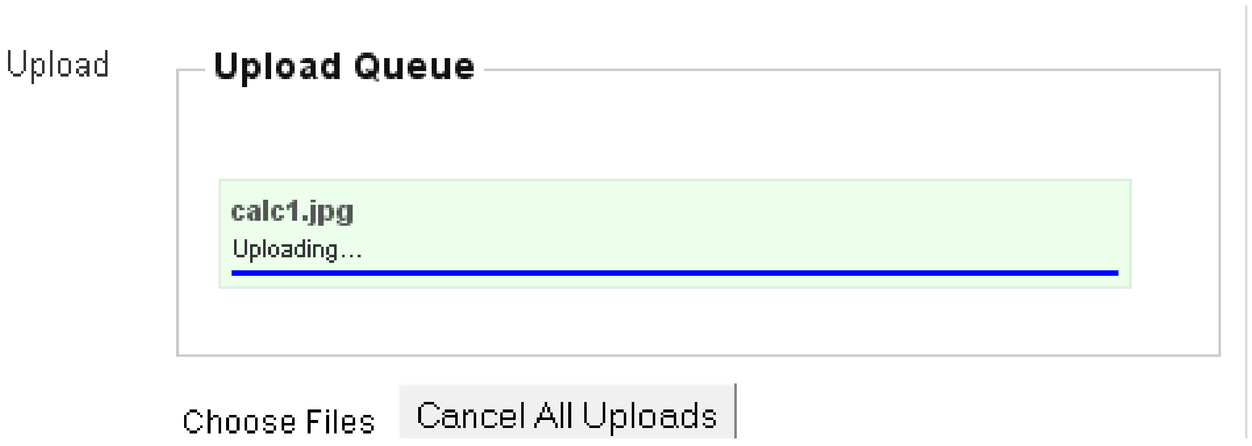


it exactly as a default option list, but it will display at your form this way:

The whole list is initialy hidden. As soon as user begins to input text, options are filtered and shown to be selected.



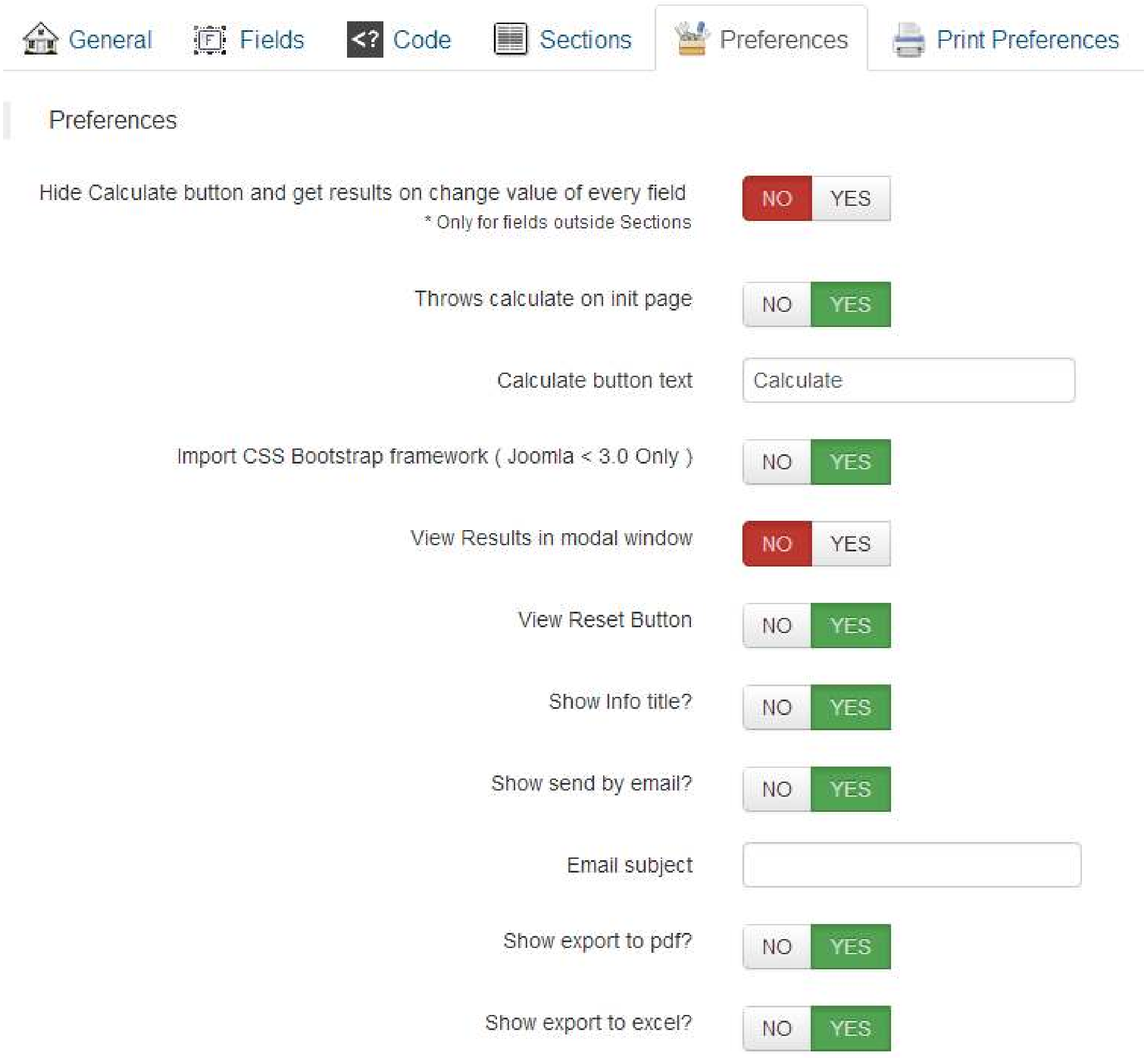
Upload file: With this input type you will allow file upload from your input form. Options to configure number of files allowed and other restrictions are set at 'Preferences tab', please check that section



below.

Preferences and print/export options

You can configure the following parameters for each of your calculators:



Preferences

Hide Calculate button and get results on change value of every field
* Only for fields outside Sections — [✓ NO] YES

Throws calculate on init page — NO [✓ YES]

Calculate button text — Calculate

Import CSS Bootstrap framework (Joomla < 3.0 Only) — NO [✓ YES]

View Results in modal window — [✓ NO] YES

View Reset Button — NO [✓ YES]

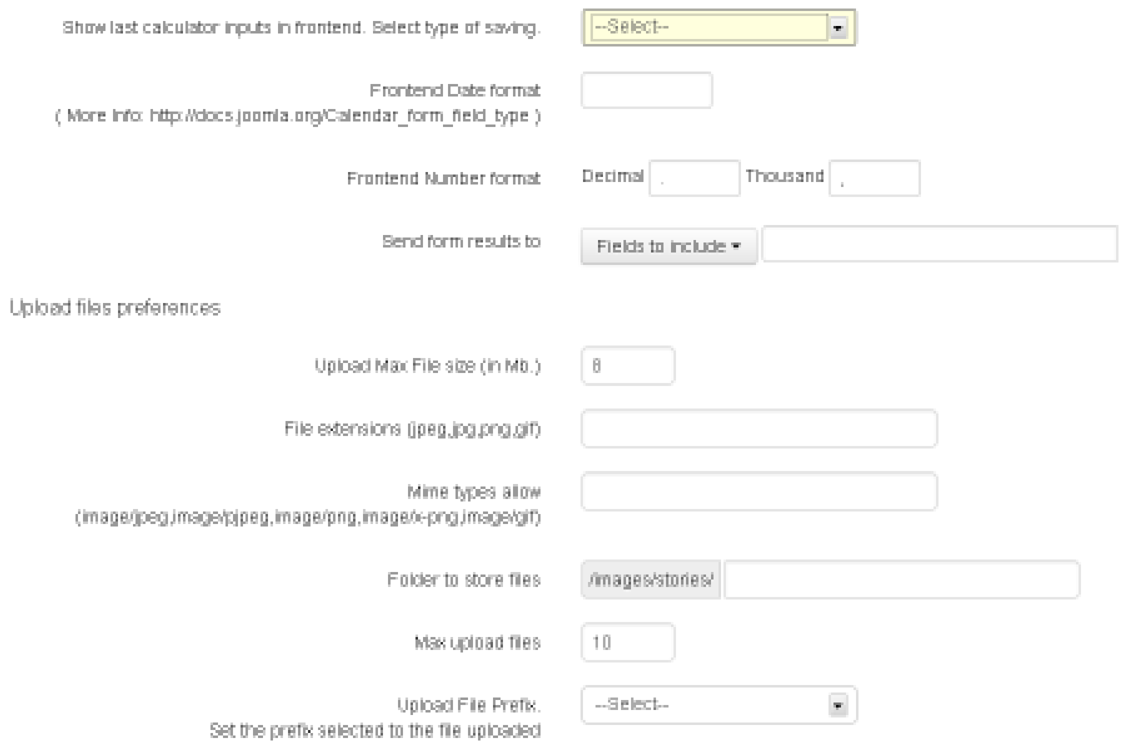
Show Info title? — NO [✓ YES]

Show send by email? — NO [✓ YES]

Email subject —

Show export to pdf? — NO [✓ YES]

Show export to excel? — NO [✓ YES]

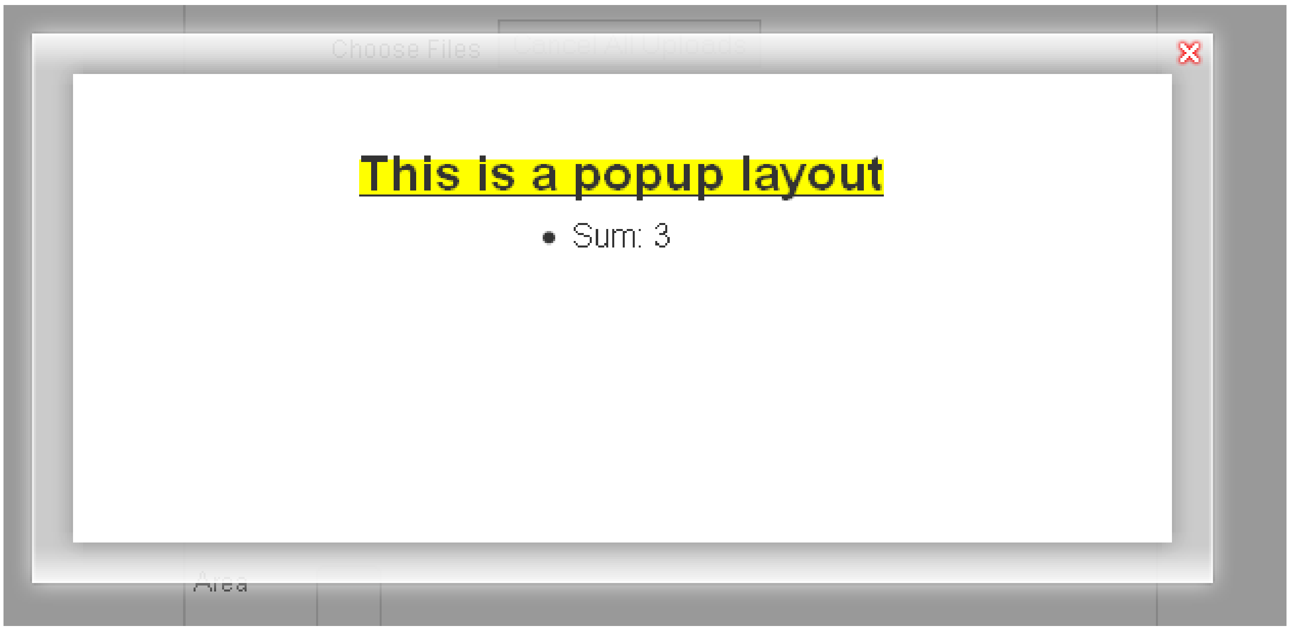


Show last calculator inputs in frontend. Select type of saving.	--Select--
Frontend Date format (More info: http://docs.joomla.org/Calendar_form_field_type)	
Frontend Number format	Decimal . Thousand ,
Send form results to	Fields to include ▾

Upload files preferences

Upload Max File size (in Mb.)	8
File extensions (jpeg,jpg,png,gif)	
Mime types allow (image/jpeg,image/pjpeg,image/png,image/x-png,image/gif)	
Folder to store files	/images/stories/
Max upload files	10
Upload File Prefix. Set the prefix selected to the file uploaded	--Select--

Hide calculate button: enable this option to hide the default calculate button. Each time an input field is changed, the calculator will lauch automatically and display the result section.
Calculate button text: label to be shown for the button 'Calculate'.
Import CSS Bootstrap: From joomla 3.0, bootstrap style classes are already loaded by default. If you are using older joomla version you can choose to import this framework, or use your default template style.
View results in modal window: Enable this option to show the result section inside a pop-up layer instead of its default position below the calculator.



This is a popup layout

- Sum: 3

View reset button/show info title/show send by email/show export to pdf/show export to excel: Enable/disable these options to hide/show different options of the calculator.

Show last calculator inputs: Select if you want to store last input data from each user attached to its joomla user or a cookie for his system. Next time he visits the calculator page, the input data will be loaded by default.

Frontend date format: Fill your preferred date format according to your location. You can check valid input formats at the page:

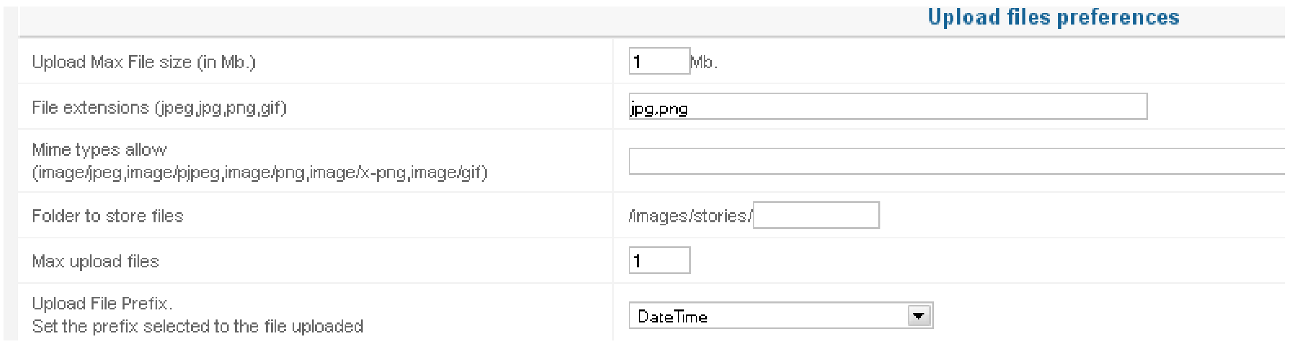
http://docs.joomla.org/Calendar_form_field_type

Front end number format: Configure your decimal and thousand separators you will allow for user input.

***Note**: these separators will be valid for user **input.** If you need to format your results in order to print the calculator output , you must use the number format function at the code section, check an example downloading the 'formatting numbers' example calculator.*

Send form results to: Place here a valid email address and everytime calculator is launched a copy containing calculator data will be sent to that email. You can also add input fields (email entered by your user at the form) to be the destination of this automatic email. Check how you can add headers/footers to email and exported files at 'print preferences' section below

Upload files preferences



Upload files preferences

Upload Max File size (in Mb.)	1 Mb.
File extensions (jpeg,jpg,png,gif)	jpg,png
Mime types allow (image/jpeg,image/pjpeg,image/png,image/x-png,image/gif)	
Folder to store files	/images/stories/
Max upload files	1
Upload File Prefix. Set the prefix selected to the file uploaded	DateTime

You should fill this configuration section if you plan to include an upload file field type to your calculator form. Here you can set:

Upload Max size: Max size allowed for each file uploaded (in Mb)

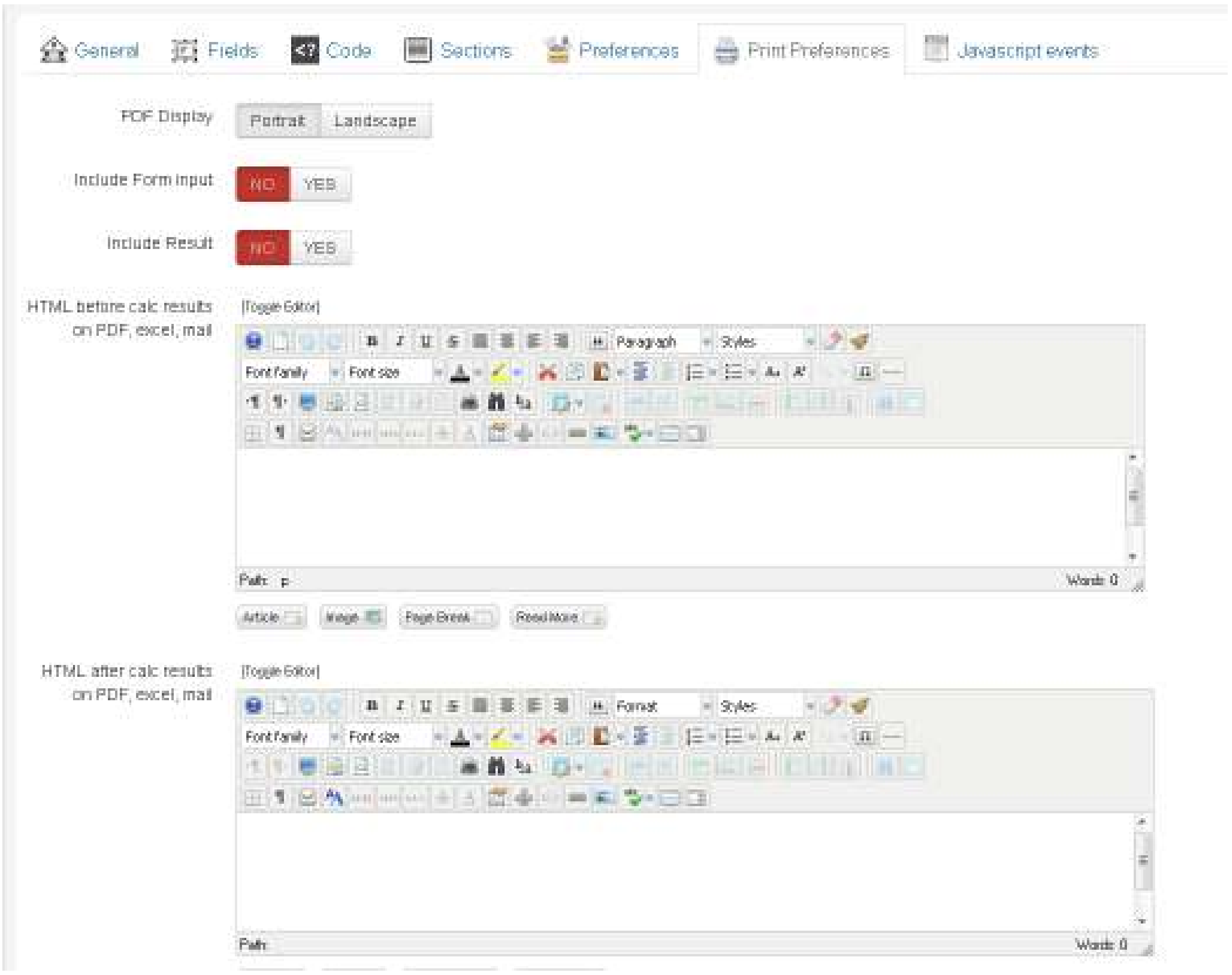
Note: This restriction will work always under your server settings. If you have set at your server limits a maximum upload size of 5 Mb, you can set here 1 or 3 Mb, but never 10Mb.

File extensions/file mimetypes: Set restrictions (comma separated) of the file types or mimetypes you will allow to upload.

Max upload files: The upload field allows several files selection, limit number of files to be uploaded in a single form submission setting this parameter.

Upload file preffix: Select a unique identifier preffix for your files (datetime and/or user id, calculator id..) to avoid files overwritten when uploaded with same name.

Print preferences



PDF Display: Portrait [✓ Landscape]

Include Form input: [✓ NO] YES

Include Result: [✓ NO] YES

HTML before calc results on PDF, excel, mail

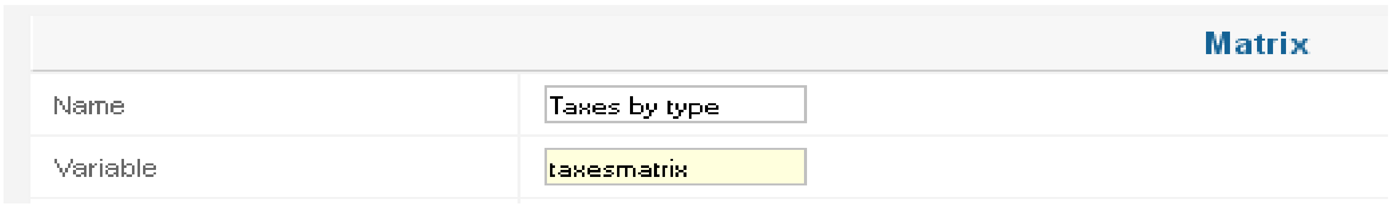
HTML after calc results on PDF, excel, mail

Using this tab you can select exported pdf files orientation (portrait/landscape), and choose if you want to include the input and/or the output of the calculator for the document. Below you have available two extra html sections to include headers and footers. Both header and footer will be included when exporting to pdf or excel, and to email content as well.

Matrix

Sometimes you may need to store some auxiliary data that you need to use at your code section. You can store data in form of tables (matrix) that you can recover at any point of your calculation. In order to create a new matrix, select 'matrix' option of your calculator, and select 'new' .

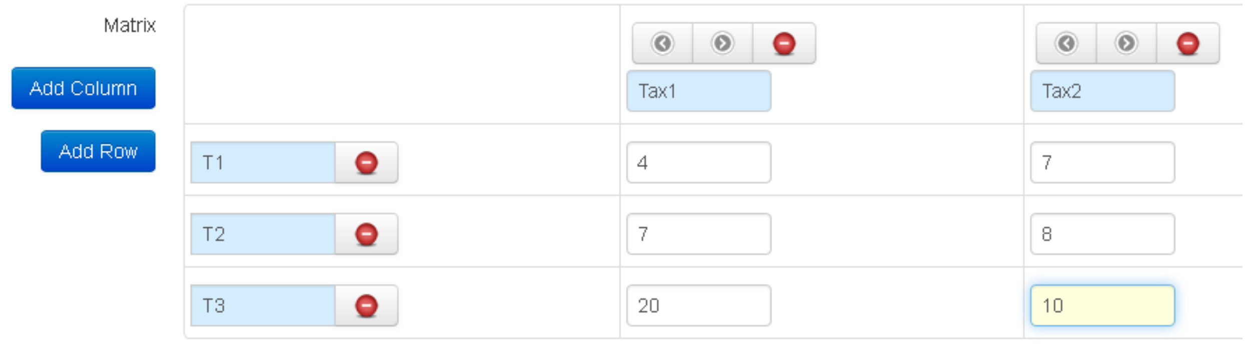
You must specify a name for your matrix, and a variable to be used at the code.



Matrix

Name	Taxes by type
Variable	taxesmatrix

You can enter the table data adding columns and rows manually or using csv data. We'll create an example with a small table, that will store two taxes values (rows), depending on a product type we have configured as input field using an option list with values (T1,T2 and T3)



	Tax1	Tax2
T1	4	7
T2	7	8
T3	20	10

Matrix data is accessed using its variable name and the row-column combination to access each cell, using this sintax:

$*matrixname*["*row*"]["*column*"]

For this example, if we write the code:

```
$result= $taxesmatrix[“T2”][“Tax2”];
```

We’ll have the value “8” at ‘result’ variable.

If we have already asked the user to choose from different product types, and we have the value at $option variable, we could use:

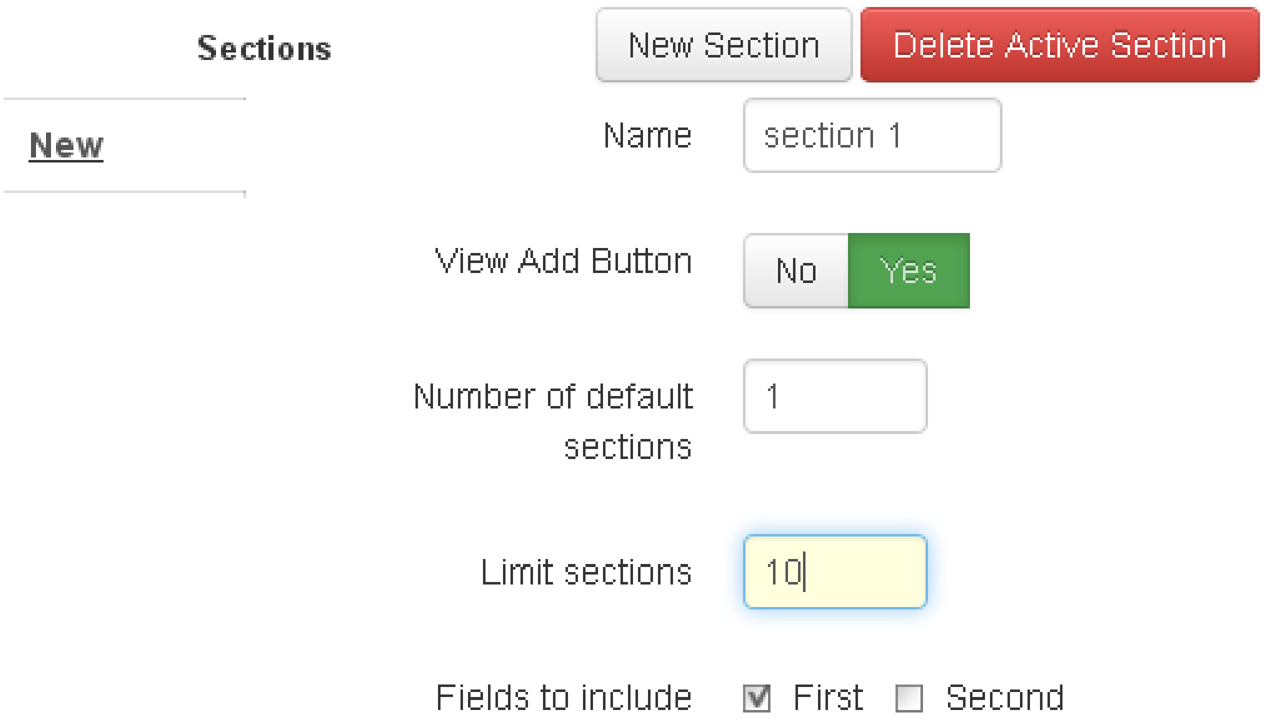
```
$result= $taxesmatrix[$option][“Tax1”];
```
Or
```
$result= $taxesmatrix[$option][“Tax2”];
```

In order to access each column cell.

You can create as many matrix as needed to store auxiliary data you need for your calculations.

Sections

A section is a part of the form that we want the user to add several times with the same structure. Each section contains its own layout and set of fields included.



Name: section 1

View Add Button: No [✓ Yes]

Number of default sections: 1

Limit sections: 10

Fields to include: [✓ First] Second

To configure a new section go to 'sections' tab of your calculator, create a new section and fill in the following data:

Name: This is the identifier of the section.

View add button: Choose if you will allow user to add new blocks.

Number of default sections: Number of blocks that will be shown by default when form is created.

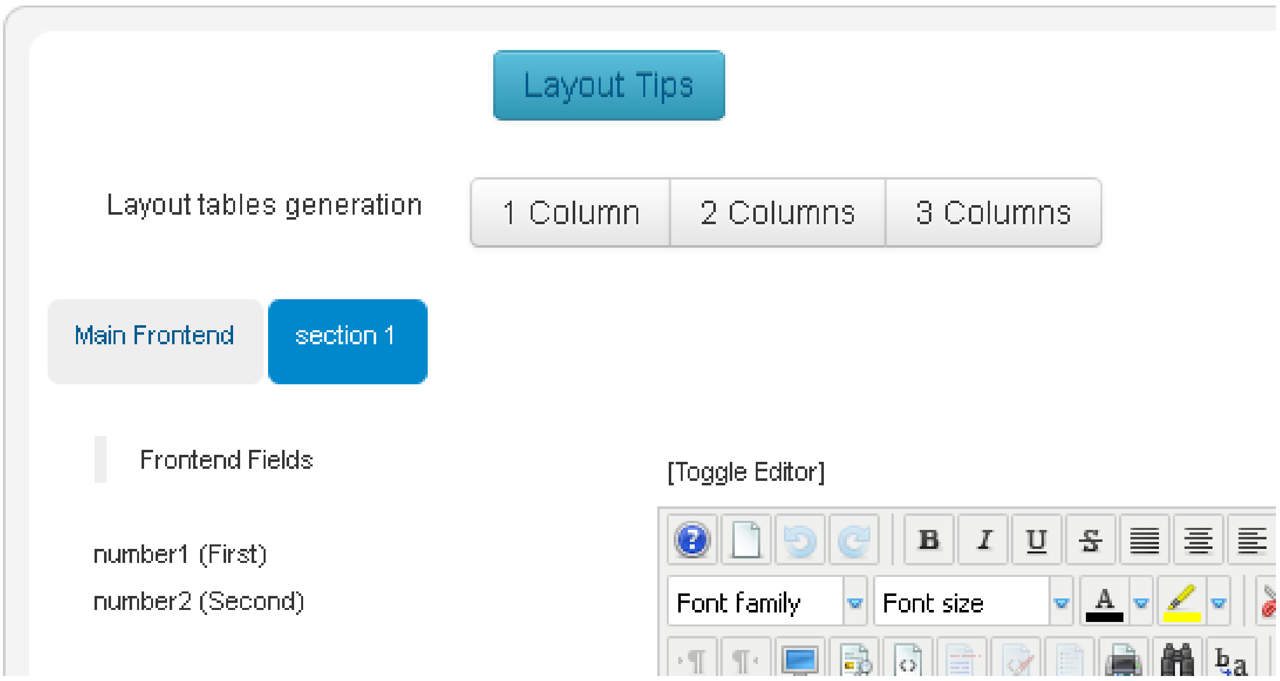
Limit section: Maximum number of blocks user can add to the section.

Fields to include: Select which of the fields configured at 'fields and code' section will be included for this section.

We'll create an example with two numeric fields. We want the user to be able to add as many rows as needed to enter new data. We'll show a row created by default, and will allow up to 10 rows of data. The name of the section will be 'section1'.

Sections layout.

Once the section is created, you must configure the layout of the section, and locate the section itself inside the main form of the calculator. Go to ‘form layout’, and you will see new options will show to allow you configure your main layout and the sections layout.

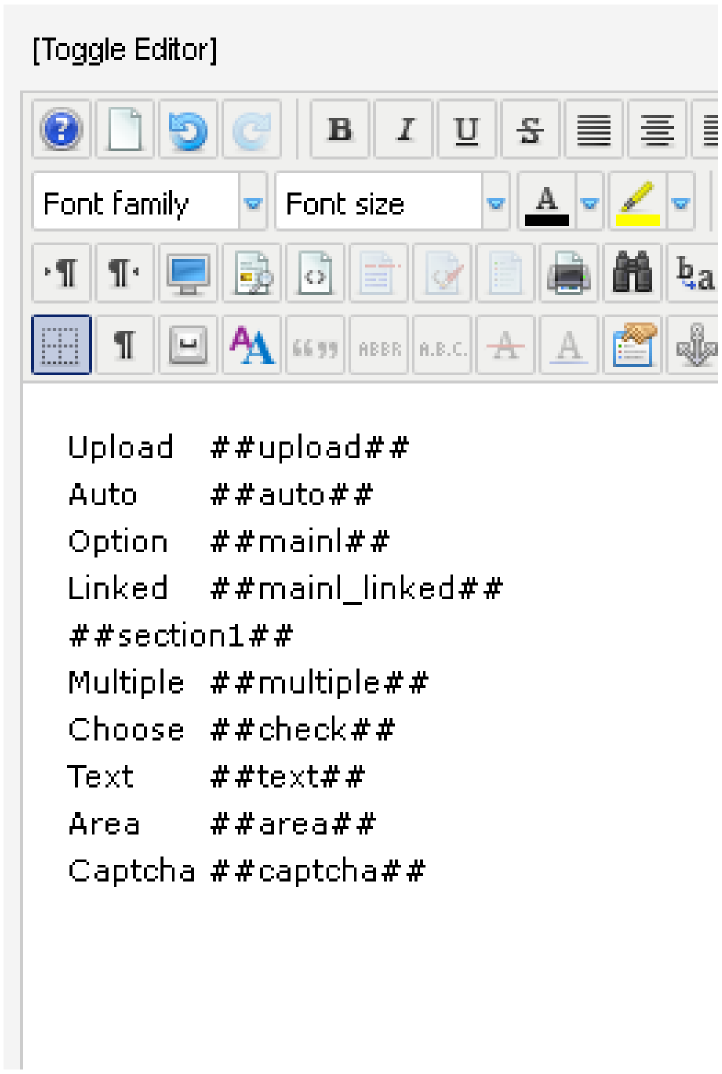


Frontend Fields

number1 (First)
number2 (Second)

Switch among the main frontend tab and your sections to configure your preferred layout for each of them. Switch among the editors clicking on their names.
At the left side you will see information about which fields belong to the main form or are assigned to the sections.
When building sections layout you can also use the layout helpers to create 1/2/3 columns layout with a single click. For this example we’ll include our two number fields in a single row.

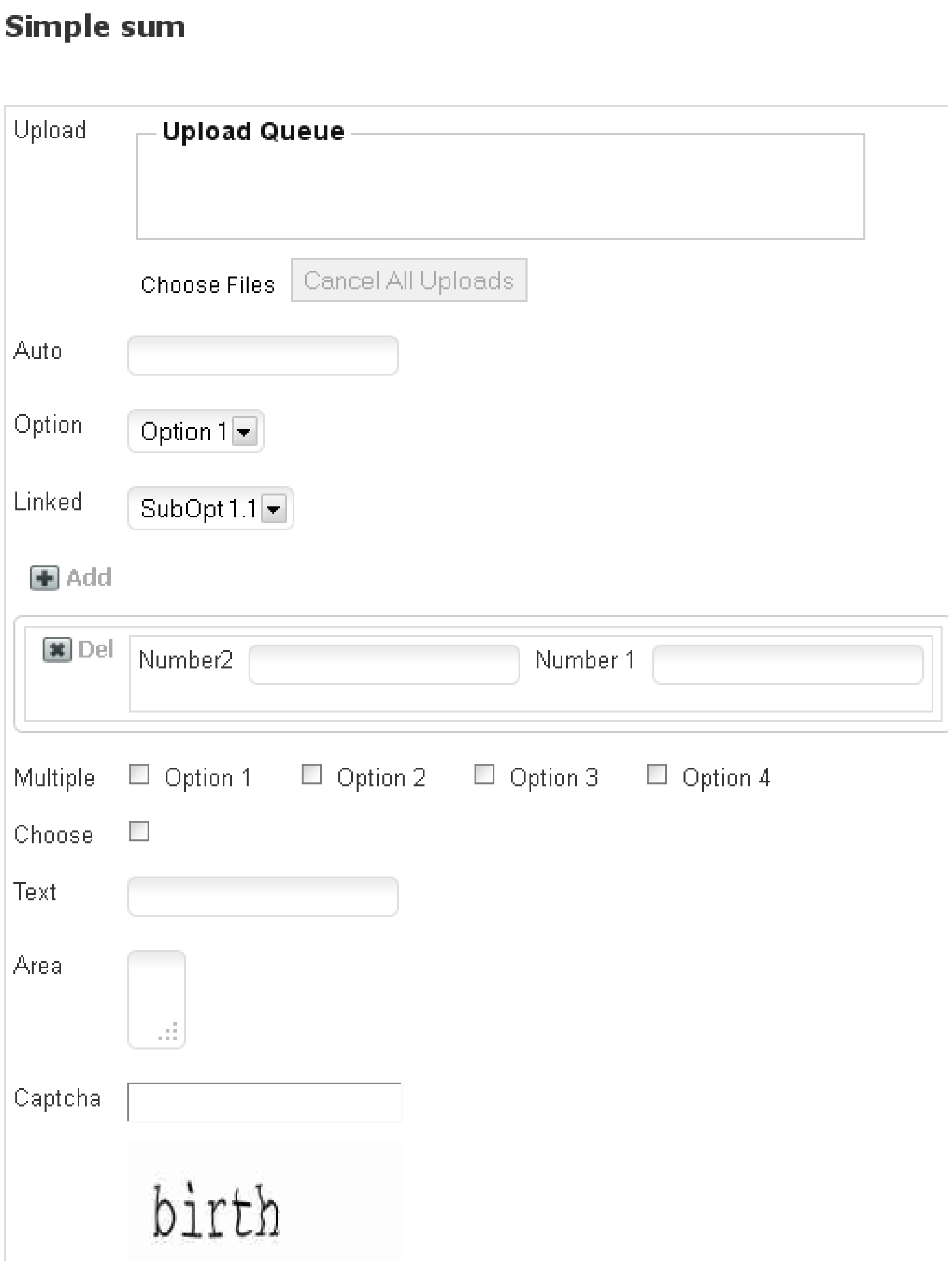
At the main layout, you must place the section as well, this time we’ll include the section in the middle of the main form fields. To include the section, change to main layout editor and include the section at the required place. Use the ##xxx## sintax same way it’s done with the other fields



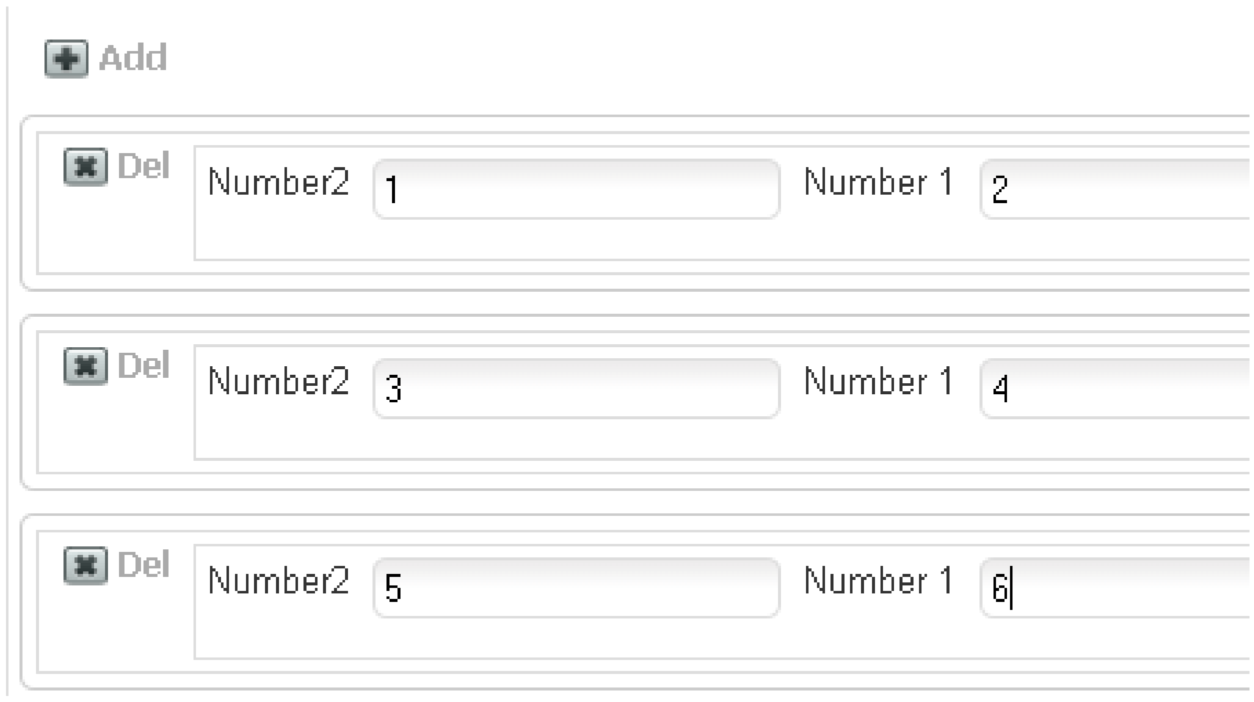
[Toggle Editor]

```
Upload   ##upload##
Auto     ##auto##
Option   ##mainl##
Linked   ##mainl_linked##
##section1##
Multiple ##multiple##
Choose   ##check##
Text     ##text##
Area     ##area##
Captcha  ##captcha##
```

This form will show at the front-end as follows:



You will note number1 and number2 fields are inside a special block where users can add or delete new blocks.



Number2	1	Number 1	2
Number2	3	Number 1	4
Number2	5	Number 1	6

***Note**: Handling the results*

When dealing with sections, the variables num1 and num2 don't have a single value anymore. You will have to use $num1 and $num2 as an array, like you do with multioption values. So $num1[0] will have the value of the first row, $num1[1] of the second row, and so on.

Javascript

If you need to include any kind of script at your form, to perform dynamic actions, you can enter your script at javascript events tab. There you can place any javascript code needed, with different sections to include the code:

Javascript events. Executed on loaded page. : Code introduced here will be executed as soon as the page finishes loading.

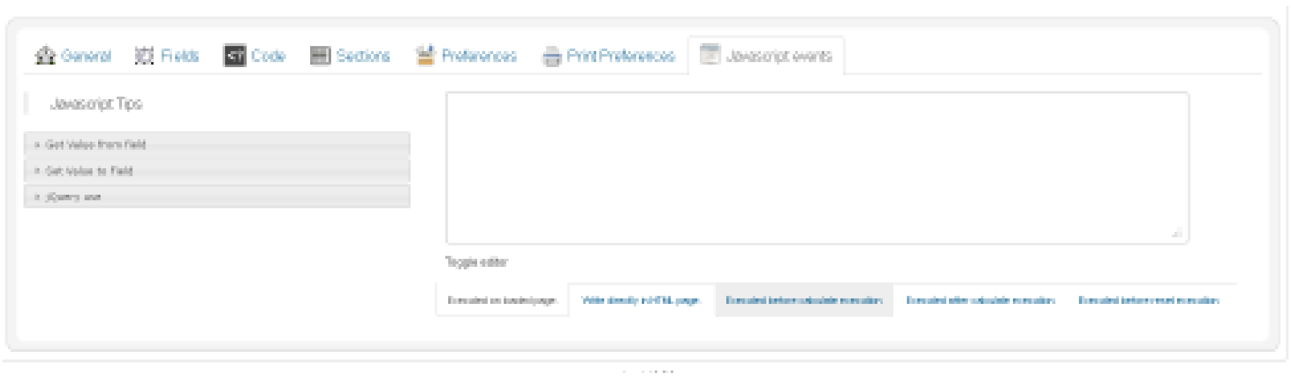
Javascript functions.: This code will be inserted inside a script tag, you can enter here functions that you will use afterwards.

Javascript executed before calculate execution: This code will be attached to the calculate button, when clicked all this code will be executed before submitting the form.

Javascript executed after calculate execution: This code will execute once the result has been displayed.

Javascript executed before reset execution: This code will be attached to the reset button, when clicked all this code will be executed before clearing the form.

You have some helpers available to handle input fields you have at your form. You will find some tips at the left panels explaining different predefined functions you can use.



Tabbed form

The module loads by default a library used to create tabs in an easy way. To create a form with several tabs user can navigate, you only have to configure a custom layout using a naming convention to identify the different tabbed sections.

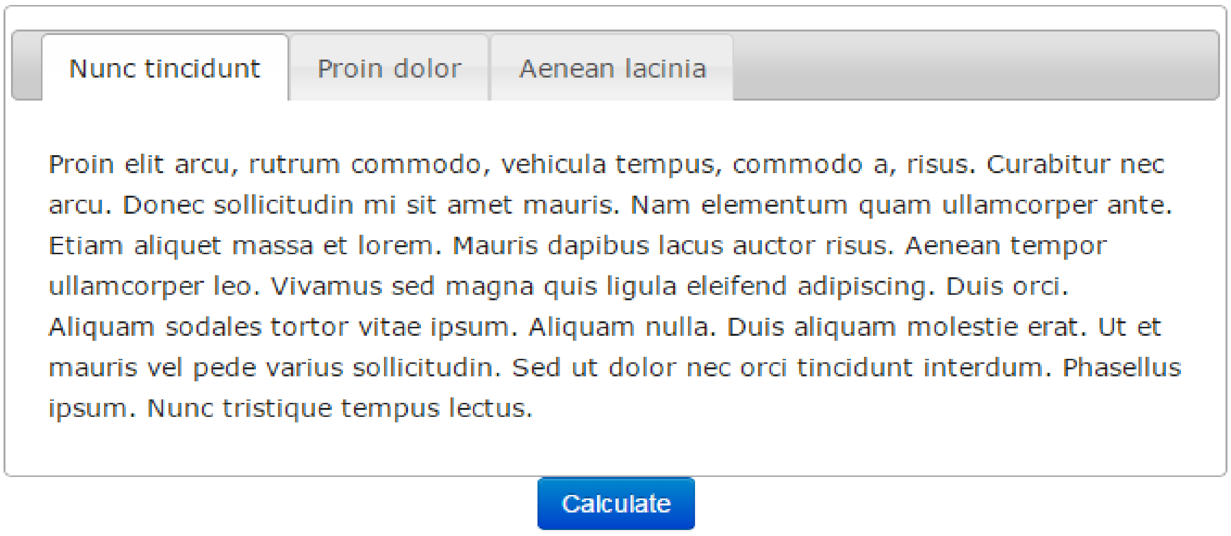
A simple html structure should look like this, you can add as many tabs as needed:

```
<div id="tabs">
<ul>
<li><a href="#tabs-1">Nunc tincidunt</a></li>
<li><a href="#tabs-2">Proin dolor</a></li>
<li><a href="#tabs-3">Aenean lacinia</a></li>
</ul>
<div id="tabs-1">
<p>Proin elit arcu, rutrum commodo, vehicula tempus, commodo a, risus. Curabitur nec arcu. Donec
sollicitudin mi sit amet mauris. Nam elementum quam ullamcorper ante. Etiam aliquet massa et lorem. </p>
</div>
<div id="tabs-2">
<p>Morbi tincidunt, dui sit amet facilisis feugiat, odio metus gravida ante, ut pharetra massa metus id nunc.
Duis scelerisque molestie turpis. </p>
</div>
<div id="tabs-3">
<p>Mauris eleifend est et turpis. Duis id erat. Suspendisse potenti. </p>
<p>Duis cursus. Maecenas ligula eros, blandit nec, pharetra at, semper at, magna.</p>
</div>
</div>
```

To activate the tabs, you should include this code at the javascript section, using the first option Javascript events.

```
CB( "#tabs" ).tabs();
```

This example will show the following output at the main form:

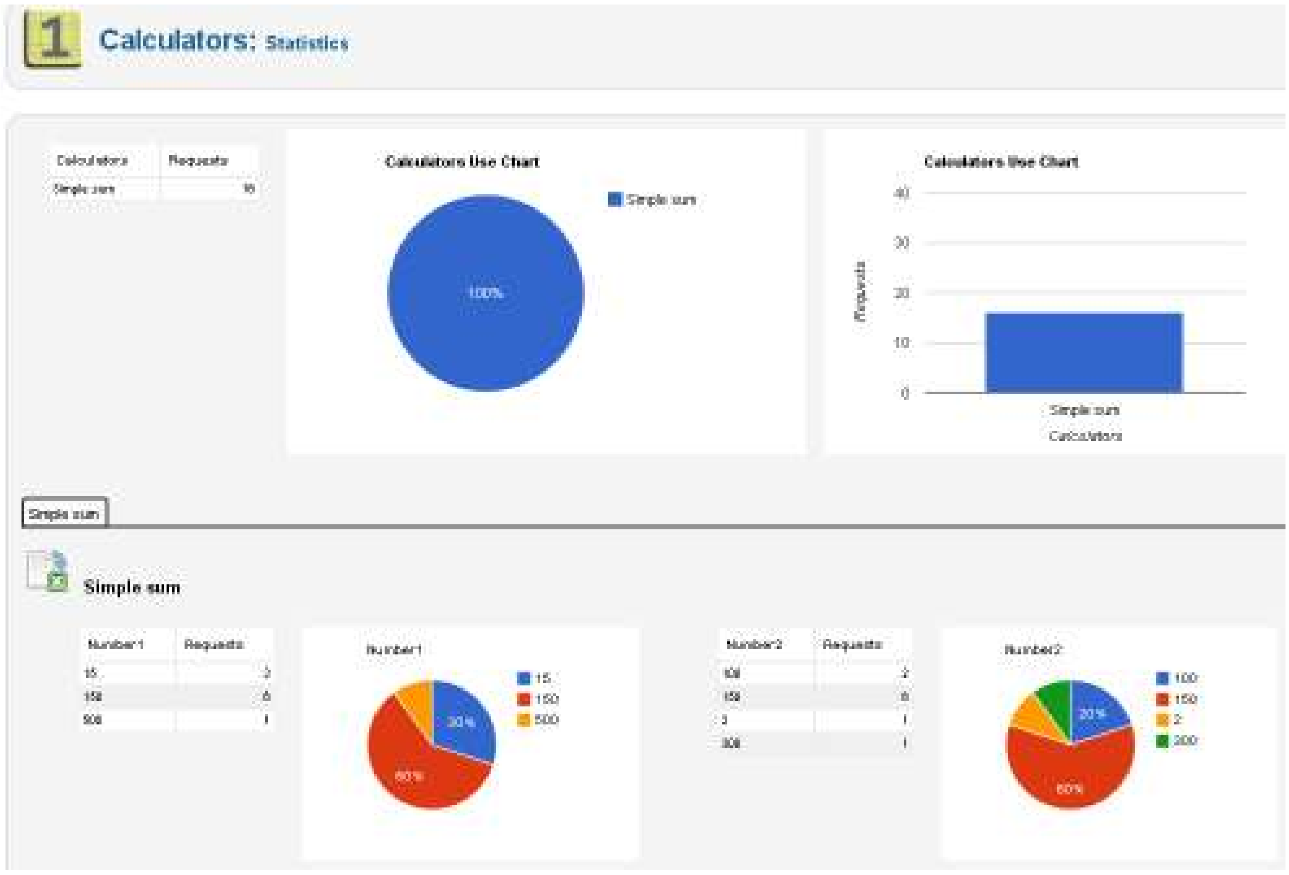


Proin elit arcu, rutrum commodo, vehicula tempus, commodo a, risus. Curabitur nec arcu. Donec sollicitudin mi sit amet mauris. Nam elementum quam ullamcorper ante. Etiam aliquet massa et lorem. Mauris dapibus lacus auctor risus. Aenean tempor ullamcorper leo. Vivamus sed magna quis ligula eleifend adipiscing. Duis orci. Aliquam sodales tortor vitae ipsum. Aliquam nulla. Duis aliquam molestie erat. Ut et mauris vel pede varius sollicitudin. Sed ut dolor nec orci tincidunt interdum. Phasellus ipsum. Nunc tristique tempus lectus.

You can place any content inside the tabs as you would normally do for default layout.

Statistics

At statistics page you will find different charts create with use data of the calculators on your site.

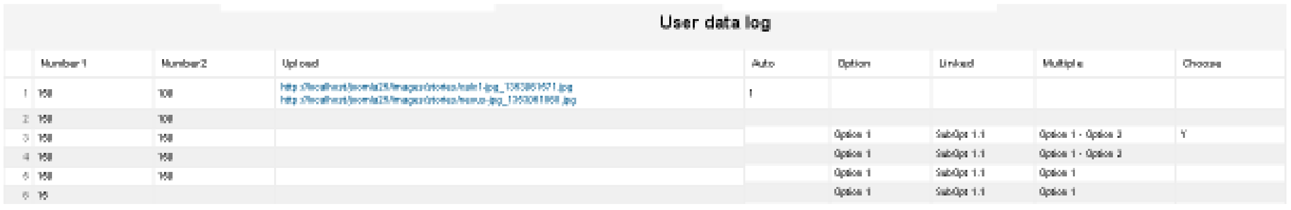


Calculators	Requests
Simple sum	16

Number1	Requests
15	3
150	6
500	1

Number2	Requests
100	2
150	6
2	1
200	1

You will find the number of requests your calculators are receiving, the distribution of the inputs, a log table with each submission that you can export to csv.



User data log

	Number1	Number2	Upload	Auto	Option	Linked	Multiple	Choose
1	150	100	http://localhost/joomla25/images/stories/calc1-jpg_1353061571.jpg http://localhost/joomla25/images/stories/menu-jpg_1353061958.jpg	1				
2	150	100						
3	150	150			Option 1	SubOpt 1.1	Option 1 - Option 3	Y
4	150	150			Option 1	SubOpt 1.1	Option 1 - Option 2	
5	150	150			Option 1	SubOpt 1.1	Option 1	
6	15				Option 1	SubOpt 1.1	Option 1	

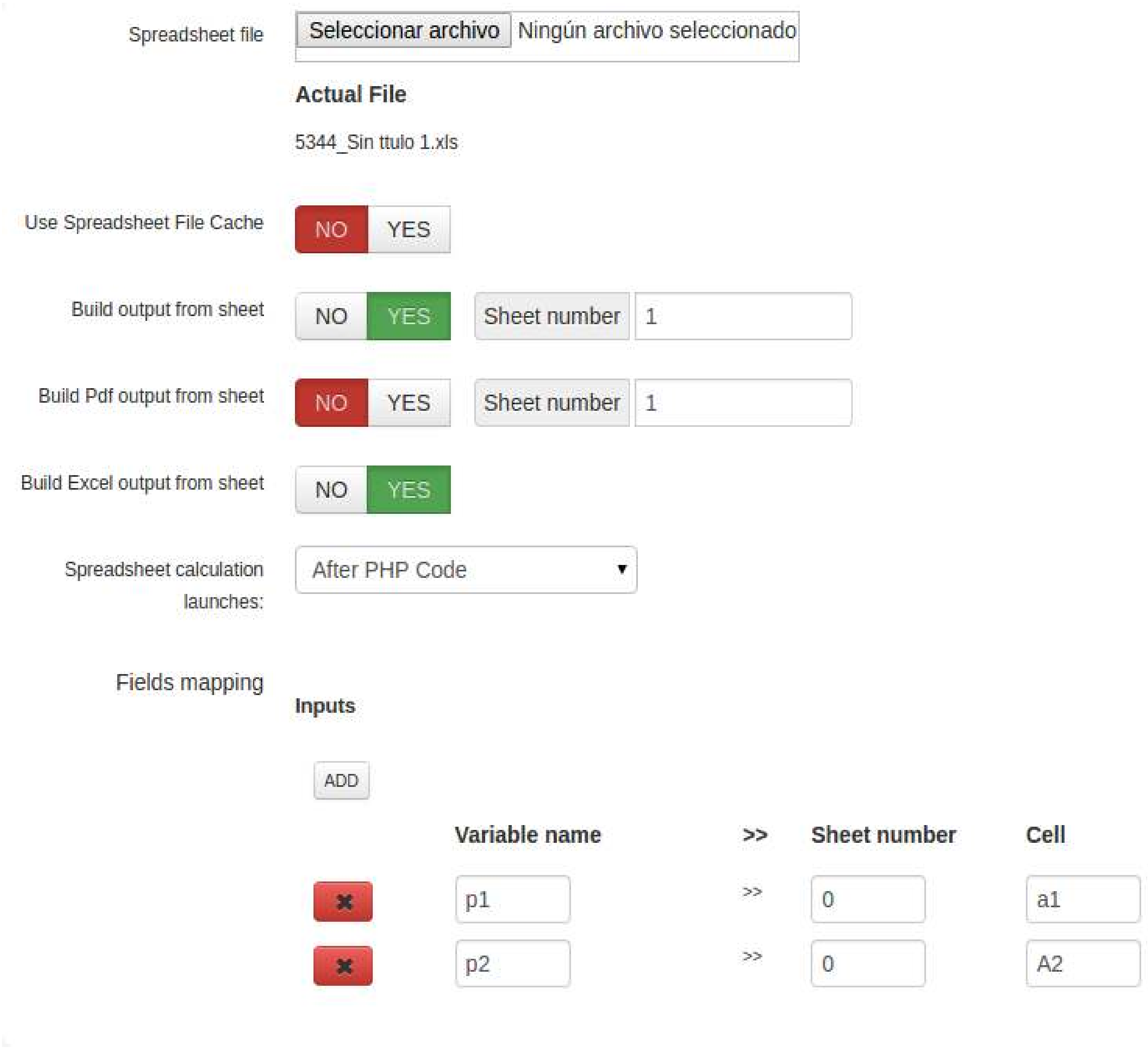
How to build your calculator from Excel

(Feature available only for CalcBuilder extended version).

If you already have an excel with all your formulas of your calculator and you need to convert it to a working form on your site, you can use the excel mapping function of CalcBuilder extended version to build your calculator in a few simple steps:

- Create a new calculator and add your input fields and input form as you will do for default calculators.
- Upload your excel using 'Spreadsheet' menu option, valid formats are .xls, .xlsx and .odt
- Map your form inputs with the cells of your excel at the 'input' section. There you have to write the name of the field at your form, and where should they be placed at your excel, with the number of sheet and the destination cell (0 is the first sheet, 1 is the second sheet...etc).
- Select one sheet as your ouput, this will discard default exit layout.
- **Or,** map your output cells to any variable. For ex, you can map cell C3 to result1.Cell D5 with result2...and now you can print your results at the output form. ##result1##, ##result2## will print the calculated cells of your excel.

You're done. Now your input form sends the inputs to the excel file, that will perform all calculations and return back the results needed.



Spreadsheet file	Seleccionar archivo Ningún archivo seleccionado
	Actual File 5344_Sin ttulo 1.xls
Use Spreadsheet File Cache	[✓ NO] YES
Build output from sheet	NO [✓ YES] Sheet number 1
Build Pdf output from sheet	[✓ NO] YES Sheet number 1
Build Excel output from sheet	NO [✓ YES]
Spreadsheet calculation launches:	After PHP Code

Fields mapping

Inputs

ADD

	Variable name	>>	Sheet number	Cell
✖	p1	>>	0	a1
✖	p2	>>	0	A2

Extra functions:

Use Spreadsheet File Cache: use this to improve performance only when you don't need to refresh your excel upload for a while. File will be cached so we suggest to activate only when you finish testing your excel sheet file.

Build output from sheet: Sheet selected will be used as output of the calculator. This will replace any exit layout you have configured using html editor.

Build PDF output from sheet: Sheet selected will be used to show contents when export to pdf is enabled. This will replace print preferences contents.

Build Excel output from file: When enabled export to excel returns your uploaded excel file filled with your inputs.

Spreadsheet calculation launches: You can also execute custom php code before and/or after the excel is launched. Use the option to select when you prefer to launch the calculations. You can also choose to execute the excel 'inside' your php code, you only need to write
//[EXCEL]
inside your code in order to launch excel calculations at that specific point. This can be used to perform initial handling of your data before sending them to excel and/or after results are recovered.

Hika Shop integration

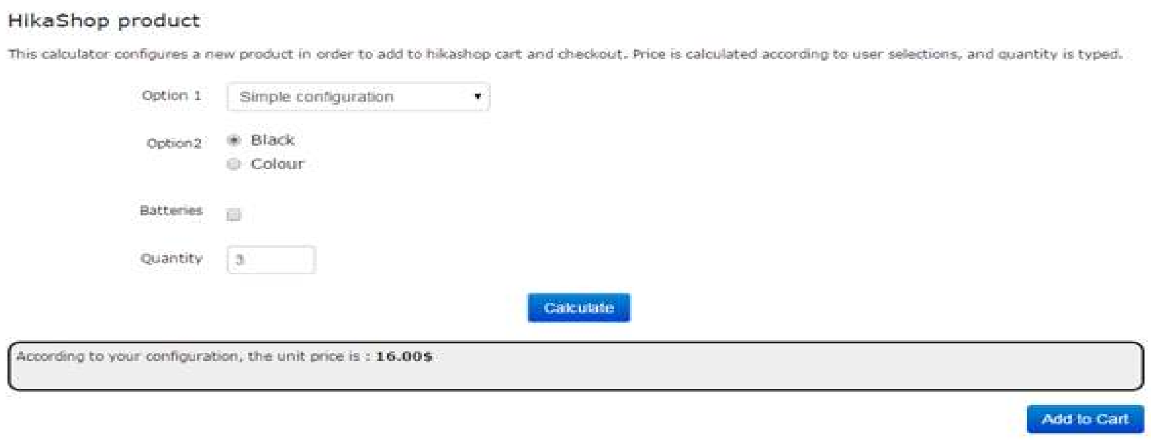
(Feature available only for CalcBuilder extended version).

In order to link the results of your calculator with hika shop cart you only need to add some extra lines at the end of your calculator code:

```
//Modify these parameters according to your product

//QTY

$hk_quantity=1;

//NAME

$hk_product_name='Test';

//LONG DESCRIPTION

$hk_product_description='Description';

//PRICE

$hk_pricehikaproduct=10;

//CODE

$hk_product_code='COD';
```

Just replacing the results in bold for your own results or calculations.

The calculator will show the results and also a new button at the bottom 'add to cart', which is in charge of creating the new product according to the configuration, add it to the cart and redirect to default checkout.



HikaShop product

This calculator configures a new product in order to add to hikashop cart and checkout. Price is calculated according to user selections, and quantity is typed.

Option 1: Simple configuration

Option2: [✓ Black] Colour

Batteries: ☐

Quantity: 3

According to your configuration, the unit price is : **16.00$**

There are extra options that you also can add to the new product. If you need to set any product attribute present at the hikashop_product table, you can add this code and fill the needed parameters:

```
//More options (from #_hikashop_product table)

$hk_productfields=array();

$hk_productfields['product_width']=23;

$hk_productfields['product_length']=12;
```

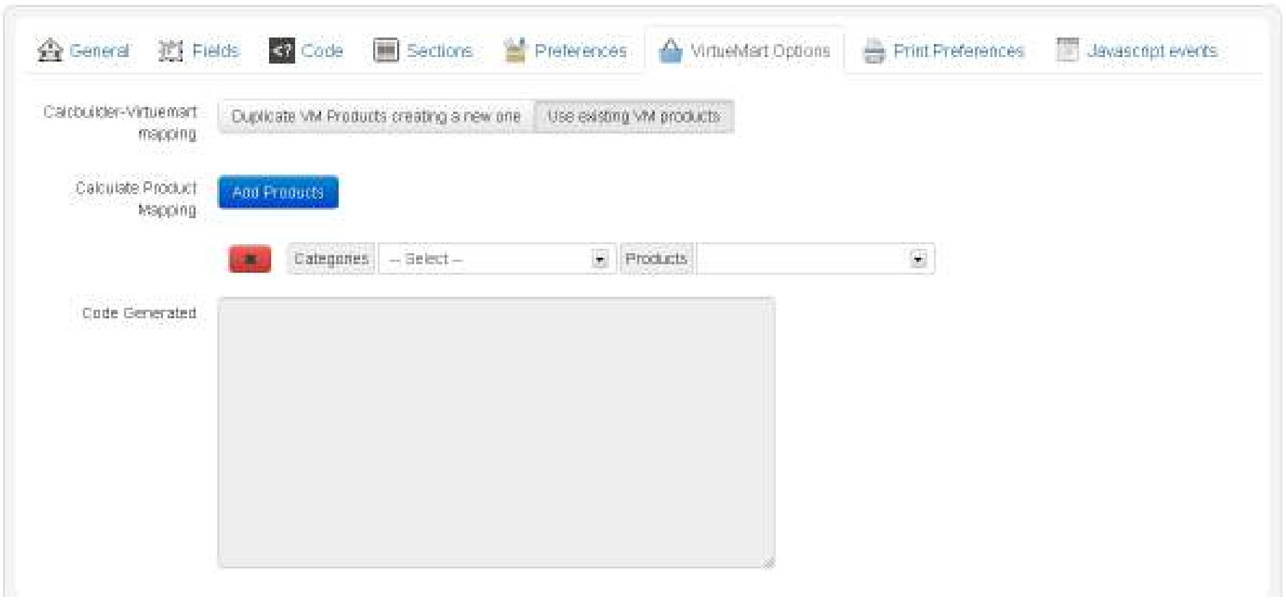
And finally, you can also attach a pdf to the new product, that will be created according to print preferences and can be downloaded once user completes payment. Only a couple of extra lines are needed to enable this function:

```
//Complete file parameters in order to attach a pdf file to your
product

$hk_file_name='PDFFile';

$hk_file_description='File Attached';
```

VirtueMart integration

(Feature available only for CalcBuilder extended version).

If you want to use your calculator results to calculate prices and/or quantities of your VirtueMart products before adding them to the cart, you need to configure the VirtueMart options tab.



Calcbuilder-Virtuemart mapping: Duplicate VM Products creating a new one | Use existing VM products

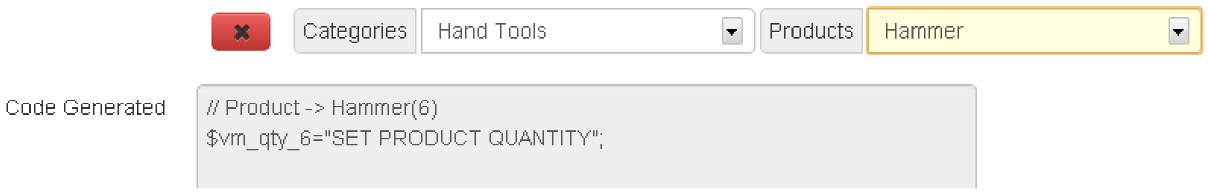
Calculate Product Mapping

Code Generated

· **Use existing VM products.**

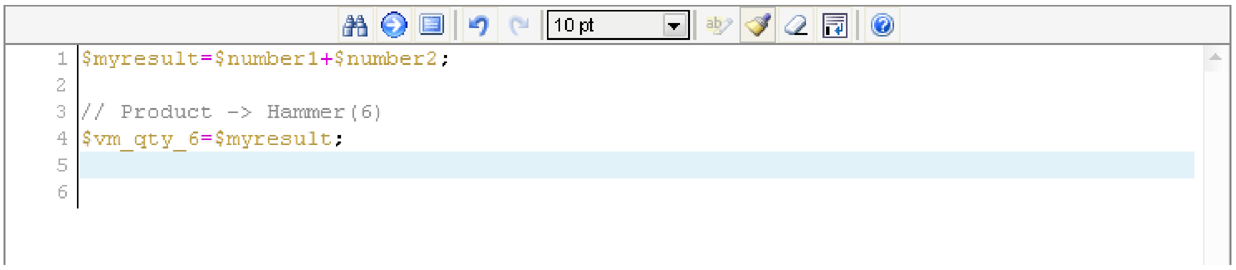
Select this option if your calculator will be used to calculate the quantity of existing products that will be added to the cart. The products are already created with a fixed price at your VM administration pages, and the calculator will get how many units of the products user needs. For ex, you can use this option to help user calculate how many paint cans he needs to paint a wall given the wall measures, how many heaters he needs for a house given the square meters of a house…everytime the product is recovered from VM, and configured with the calculated quantity before adding it to the cart.

Check ‘Use existing VM products’ and select one category and one product you want to link with the calculator output. Save and you will see the code needed to link your results with the selected product:



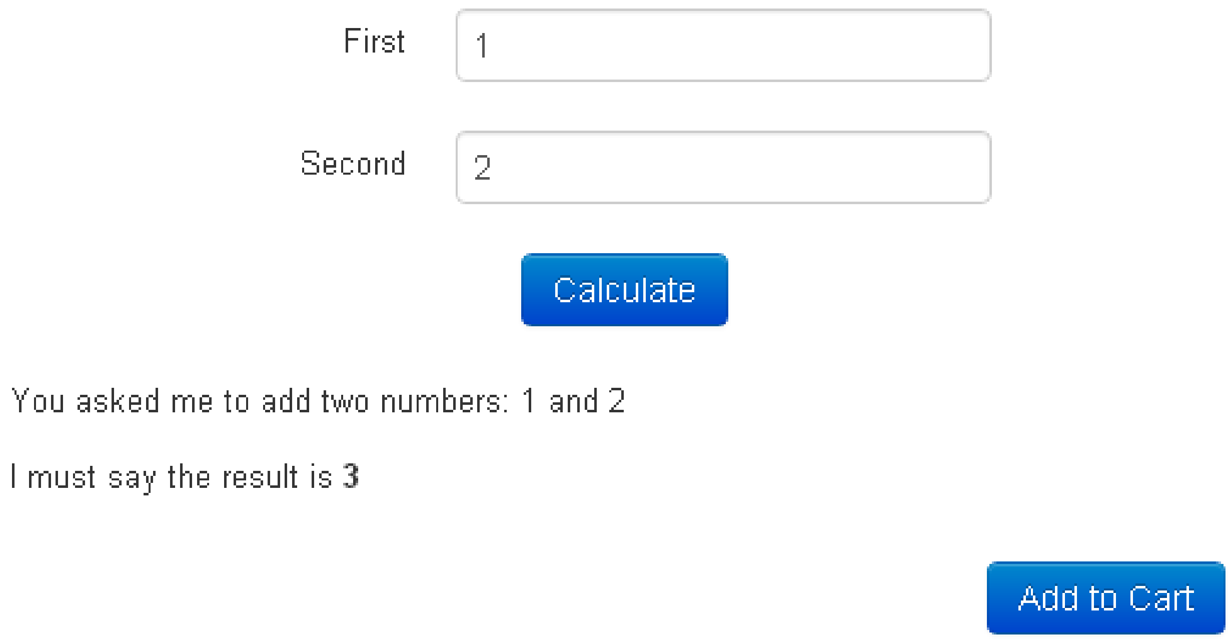
```
// Product -> Hammer(6)
$vm_qty_6="SET PRODUCT QUANTITY";
```

Copy the code generated to your calculator code section, replacing the "SET PRODUCT QUANTITY" text for your calculated quantity:



```
$myresult=$number1+$number2;

// Product -> Hammer(6)
$vm_qty_6=$myresult;
```

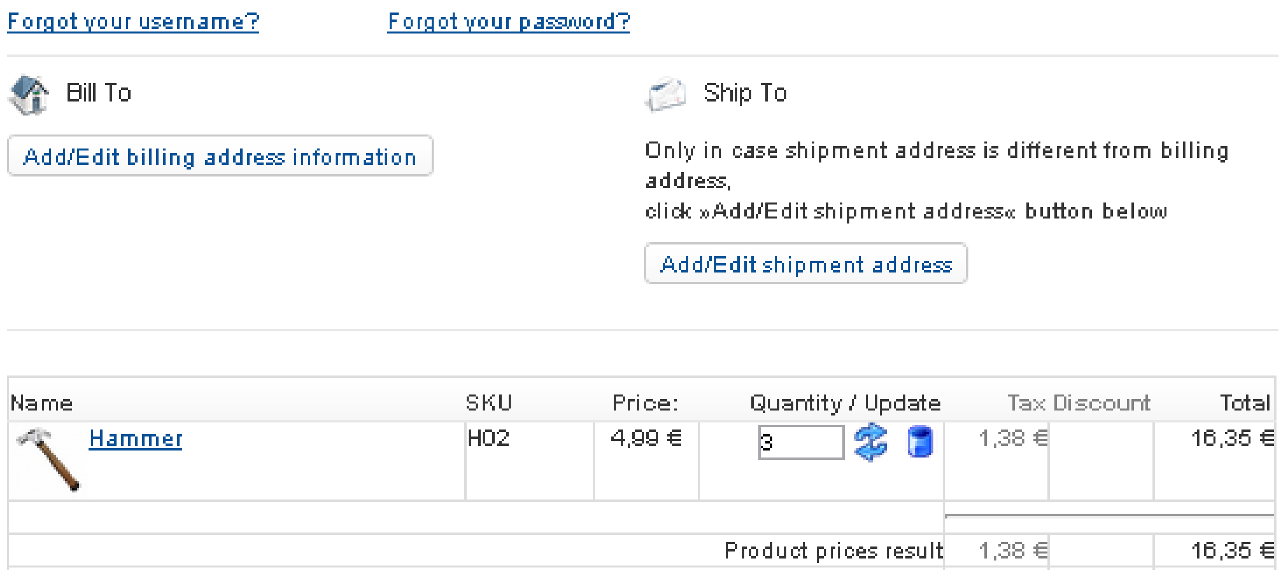
Using this example, the calculator will create an 'Add to cart' button added to the result section that we'll be in charge of adding input1+input2 hammers to your VM cart:



You asked me to add two numbers: 1 and 2

I must say the result is **3**

Add to Cart button leads to the cart view of VM, where you will see the product with the calculated quantity:



Forgot your username? Forgot your password?

Bill To

Add/Edit billing address information

Ship To

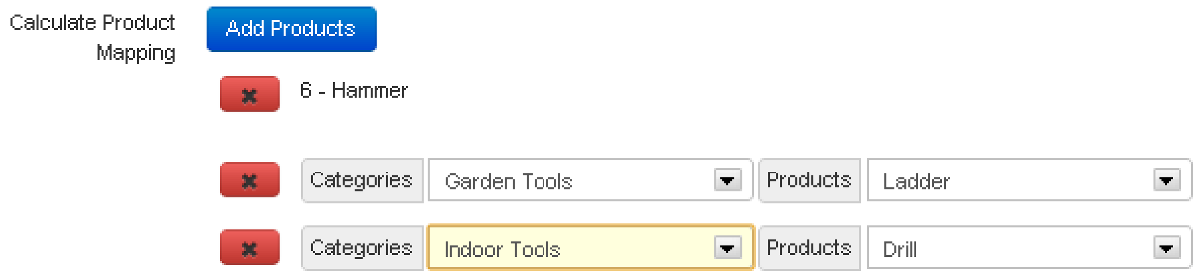
Only in case shipment address is different from billing address,
click »Add/Edit shipment address« button below

Add/Edit shipment address

Name	SKU	Price:	Quantity / Update	Tax	Discount	Total
Hammer	H02	4,99 €	3	1,38 €		16,35 €
			Product prices result	1,38 €		16,35 €

This is already your default VM view you have configured for your cart, so you can continue checkout from here.

You can add more than one product at at time, clicking ‘Add products’ to the configuration page:



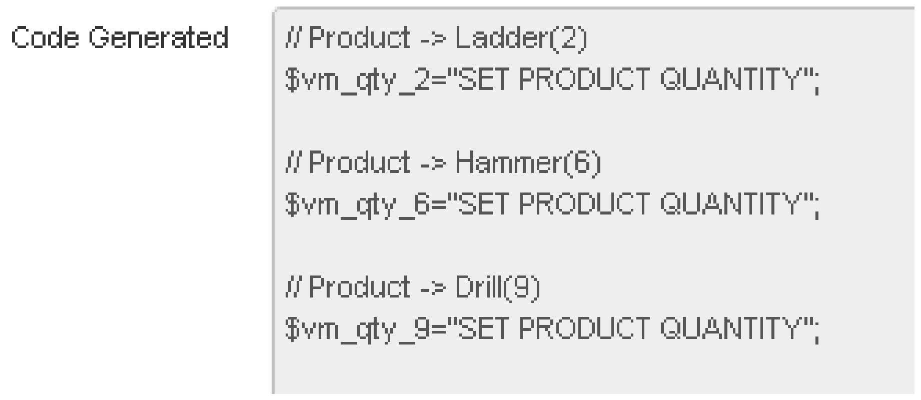
Calculate Product Mapping

Add Products

6 - Hammer

Categories	Garden Tools	Products	Ladder
Categories	Indoor Tools	Products	Drill

The code generated will contain the expressions needed to set quantities for each product at calculation time:



Code Generated

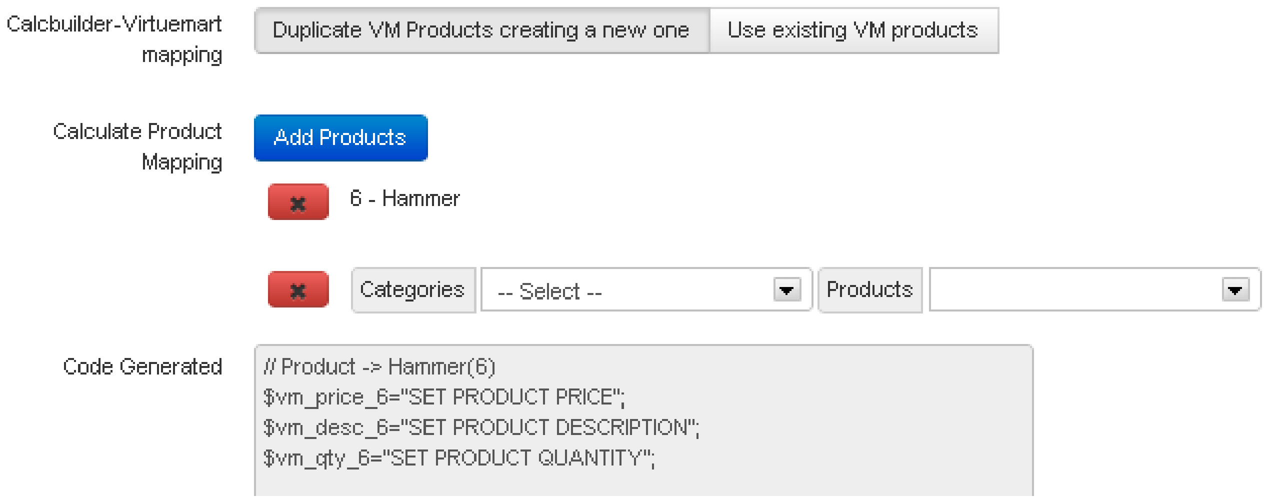
```
// Product -> Ladder(2)
$vm_qty_2="SET PRODUCT QUANTITY";

// Product -> Hammer(6)
$vm_qty_6="SET PRODUCT QUANTITY";

// Product -> Drill(9)
$vm_qty_9="SET PRODUCT QUANTITY";
```

So you could calculate and set at code tab the required values for these products quantity. If any of the products results a quantity of ‘0’ after your calculations, it will not be added to the cart.

· **Duplicate VM products**

You can not store at VM same instance of the same product attached with dynamic prices. If your calculator will set the price and/or the quantity of the products to be added, you need to choose this option. You have to create a ‘template’ of your product at VM, with all parameters you need. When calculator outputs the calculated price/quantity, the template product will be recovered, duplicated with same features, but with its new calculated price set, and will be added to the cart using the calculated quantity.

The configuration steps are the same, you must select the ‘template’ product and save to recover the code you must add to code section in order to use this product as template:


Calcbuilder-Virtuemart mapping
Duplicate VM Products creating a new one
Use existing VM products
Calculate Product Mapping
Add Products
6 - Hammer
Categories
-- Select --
Products
Code Generated

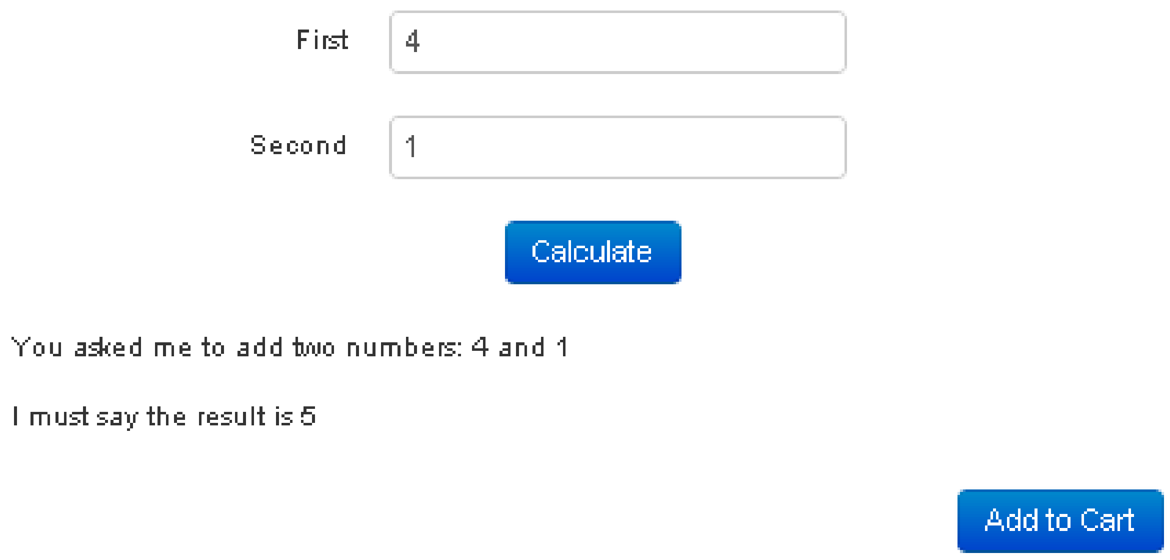
```
// Product -> Hammer(6)
$vm_price_6="SET PRODUCT PRICE";
$vm_desc_6="SET PRODUCT DESCRIPTION";
$vm_qty_6="SET PRODUCT QUANTITY";
```

This time you must fill three values with the result of your calculations. The quantity, as we did for the previous option, the price, and also the description of the product. Dynamic prices are usually due to different configuration user chooses for the product, so you can set here detailed specs of the specific instance/configuration of this product, for example:

```
// Product -> Hammer(6)
$vm_price_6=$number1+10;
$vm_desc_6="Hammer + special box";
```

$vm_qty_6=1;

Your calculator will fill dynamically the parameters and add to cart button will create the new product based on your template and add it to the cart:

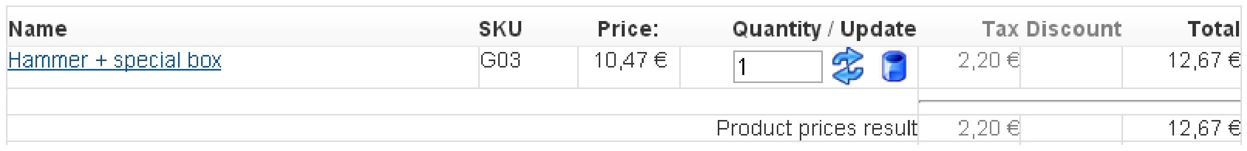


First 4

Second 1

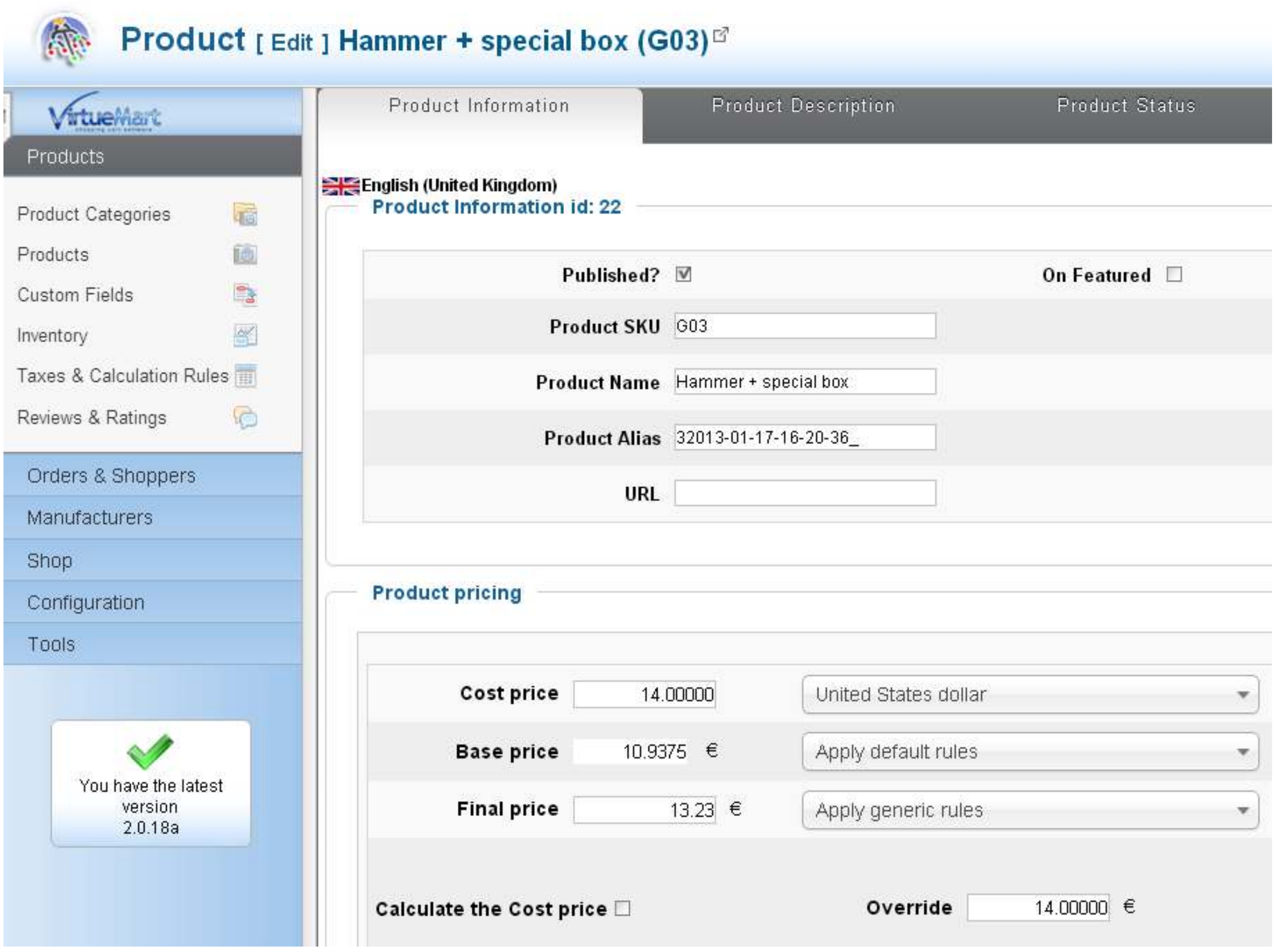
You asked me to add two numbers: 4 and 1

I must say the result is 5



Name	SKU	Price:	Quantity / Update	Tax	Discount	Total
Hammer + special box	G03	10,47 €	1	2,20 €		12,67 €
			Product prices result	2,20 €		12,67 €

If you check your VM admin you will see this new instance of the product created with the calculated price.



Product [Edit] Hammer + special box (G03)

English (United Kingdom)

Product Information id: 22

Published? ☑ On Featured ☐

Product SKU: G03

Product Name: Hammer + special box

Product Alias: 32013-01-17-16-20-36_

URL:

Product pricing

Cost price: 14.00000 United States dollar

Base price: 10.9375 € Apply default rules

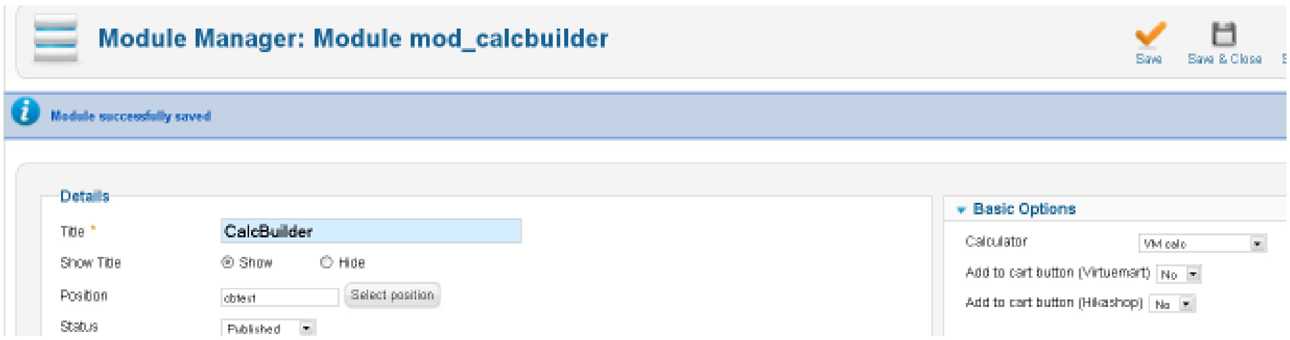
Final price: 13.23 € Apply generic rules

Calculate the Cost price ☐ Override 14.00000 €

- **Insert calculator inside existing VM product.**

With this option you will see your calculator as a custom field of your VM product, that will launch a popup with your calculator, and send the calculated price and quantity to the product page, before using the default 'add to cart button'. As the product already can have a base price, the calculator will add the calculated price to the base price of the product. The steps to configure:

- Unzip your downloaded component and install **plg_cbforvm.zip** if you're using **VM2** and **plg_cbformvm3.zip** if you're using **VM3**.
- Enable the VM Custom calculator plugin at your plugin manager.
- Create the calculator with the needed fields, and publish it using a module. It must be on state 'published', to be available at the product page, but you can select a custom position in order to keep it hidden for the rest of the site. Select 'No' for the parameters asking for showing 'add to cart button'.



Module Manager: Module mod_calcbuilder

Module successfully saved

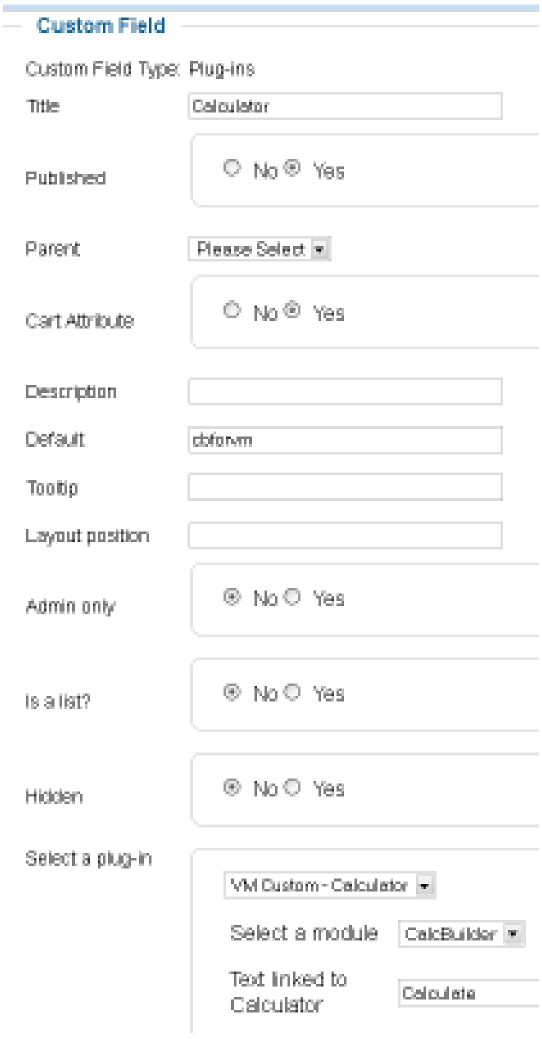
Details

Title *	CalcBuilder
Show Title	[✓ Show] Hide
Position	cbtest
Status	Published

Basic Options

Calculator	VM calc
Add to cart button (Virtuemart)	No
Add to cart button (Hikashop)	No

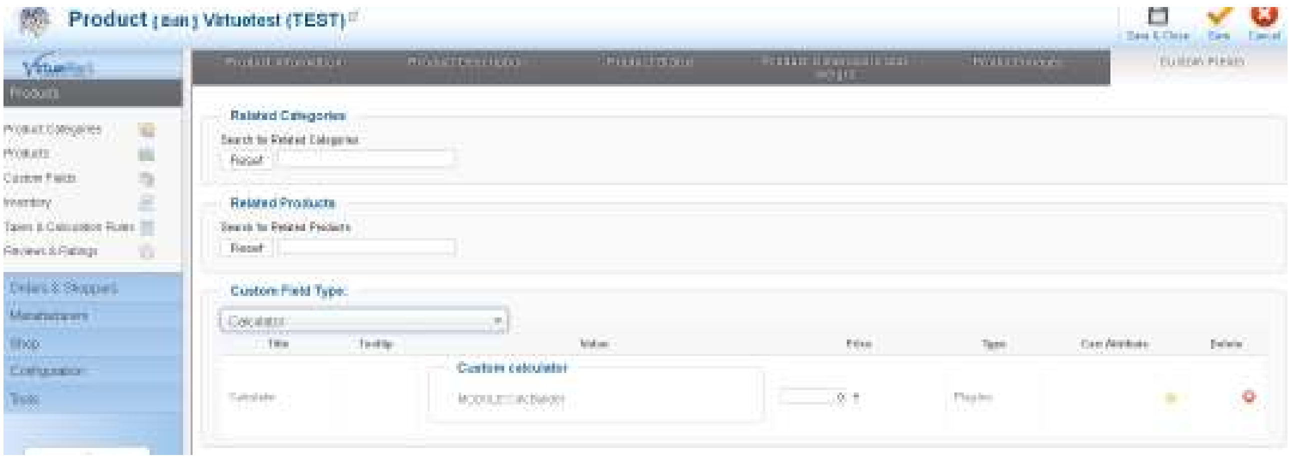
- Create a custom field at Virtuemart. Select 'plugin' as the type, ensure you have selected 'Yes' for 'published' and 'cart attribute'. Select below the VM Custom calculator plugin, and the module you created before. It will also ask you for the text that will display the link to launch calculator.



Custom Field

Custom Field Type:	Plug-ins
Title	Calculator
Published	No [✓ Yes]
Parent	Please Select
Cart Attribute	No [✓ Yes]
Description	
Default	cbforvm
Tooltip	
Layout position	
Admin only	[✓ No] Yes
Is a list?	[✓ No] Yes
Hidden	[✓ No] Yes
Select a plug-in	VM Custom - Calculator
Select a module	CalcBuilder
Text linked to Calculator	Calculate

- Add the new custom field to your VM product



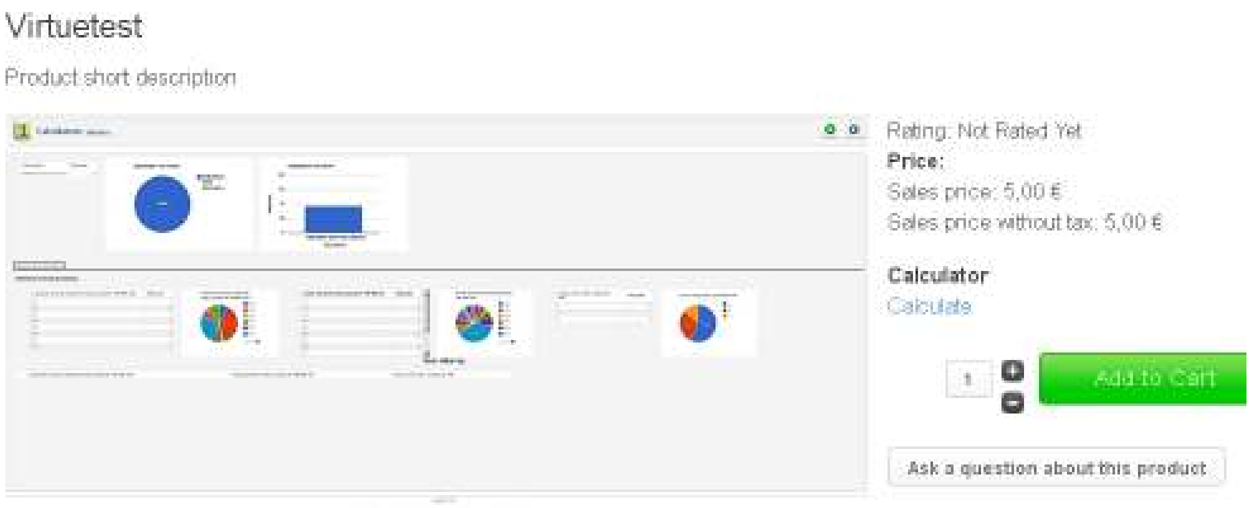
- Write the code at your calculator to get your calculated quantity and price. Then use this code to pass values to VM product:

```
//calculator.Write/calculate here your custom price, quantity, and any //additional description you need to add to your product according to the input selected.
$price=xxx;
$dsc="yyy"
$qty=zzz;

//fixed code
$session = JFactory::getSession();
$idproduct=$session->get('activevmproduct');
$session->set('calculatedqty_'.$idproduct,$qty);
$session->set('calculateddesc_'.$idproduct,$dsc);
$session->set('calculatedprice_'.$idproduct,$price);
```

For ex, we have created a simple calculator with a option list, when first option is selected, we'll add 10€ to the base price, 15€ for the second one. The code:

```
//calculator
$price=$optioncustom;
$dsc="Additional description:".$optioncustom_name;
$qty=3;
//fixed code
$session = JFactory::getSession();
$idproduct=$session->get('activevmproduct');
$session->set('calculatedqty_'.$idproduct,$qty);
$session->set('calculateddesc_'.$idproduct,$dsc);
$session->set('calculatedprice_'.$idproduct,$price);
```

So when we visit our product, a new field 'calculate' shows a link to the calculator:


Virtuetest

Product short description

Rating: Not Rated Yet

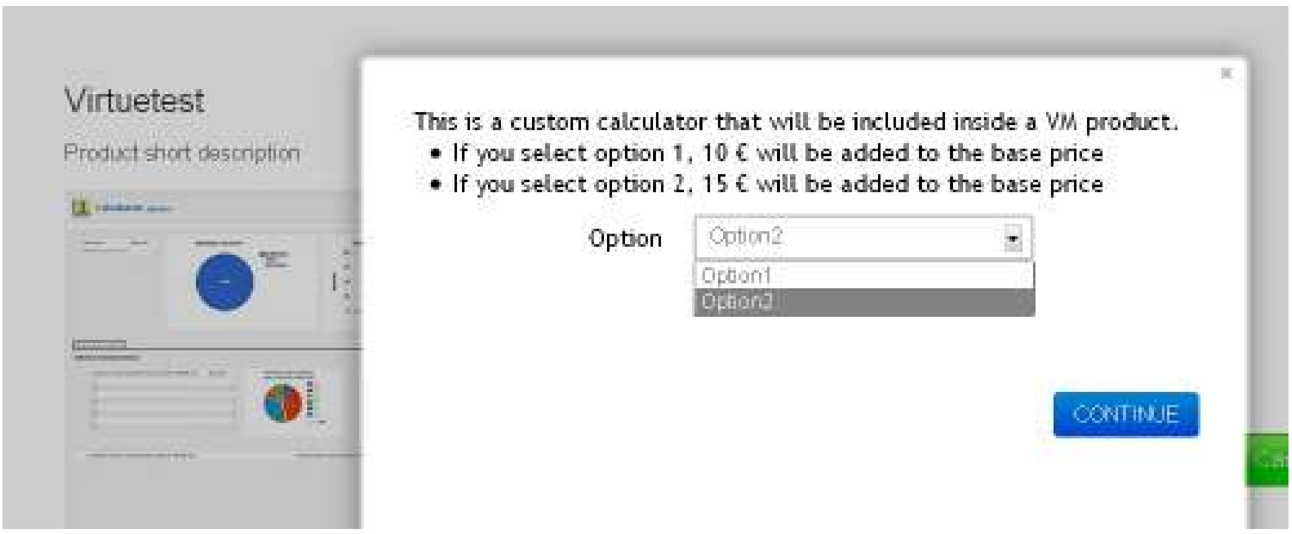
Price:

Sales price: 5,00 €

Sales price without tax: 5,00 €

Calculator

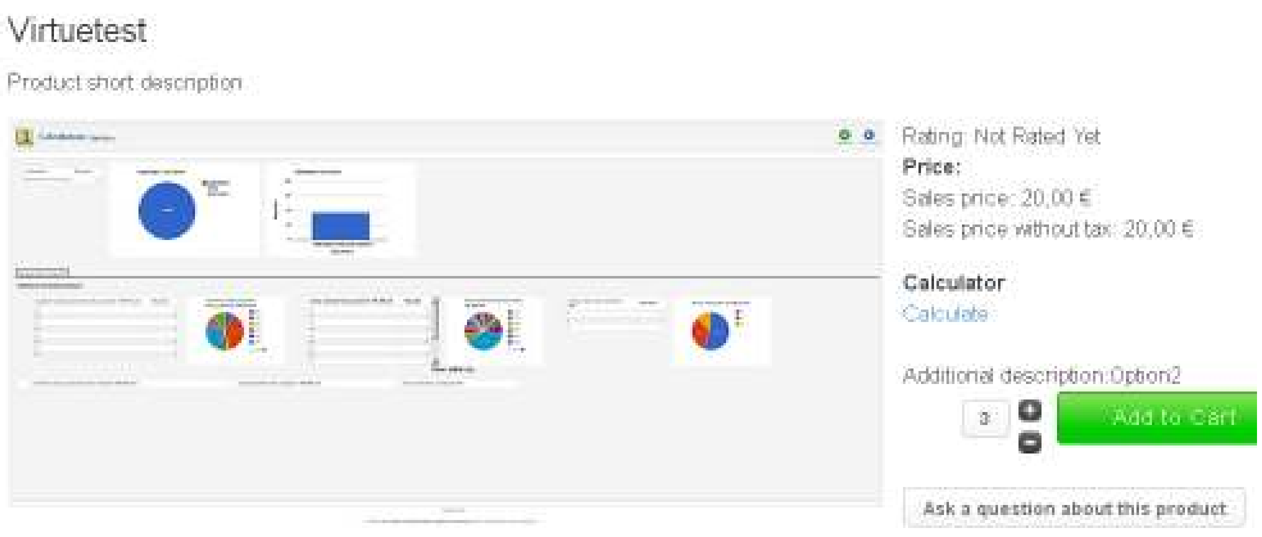
Calculate


This is a custom calculator that will be included inside a VM product.

- If you select option 1, 10 € will be added to the base price
- If you select option 2, 15 € will be added to the base price

Option: Option2

When continue is pressed, calculated values are added to the product main page:



Virtuetest

Product short description

Rating: Not Rated Yet

Price:

Sales price: 20,00 €

Sales price without tax: 20,00 €

Calculator

Calculate

Additional description:Option2

And now the checkout can continue.

Want to know how to build more advanced calculators? You can get more examples to import with your calcbuilder at your customer area. Some of them are:

- **Amortization calc**: It shows as rows as years of amortization. An example of dynamic output.
- **Body Mass Index**: Show your wellness in text depending on calculation result.
- **Checksum**: Calculate the sum of the input checked. Example of Yes/No input use.
- **Estimated Bussiness card**: Example of calculated cost based on options, in this example about making bussiness cards.
- **Formatting numbers**: Example of how to format numbers in code PHP.
- **Linked List**: Example of list with options depending on the value of another list. You can link as many lists as you need, only following the naming rules.
- **Values**: All types of input and how to get their values. Remember, fields in sections are always arrays in code.
- **Using non-output matrix:** Using matrix option is a good idea to search values. This example shows you how to use it.
- **Ship costs**: Create sections and use javascript at your form.

As you can see, Calcbuilder has a lot of features:

- Create dynamic sections (with add and del buttons) to use in forms.
- Write your own javascript code and use your created fields to add events.
- Use of matrix tables to get data easily.
- Create select fields using SQL query.
- Customized PDF and Excel prints
- Send email when user use calculator with it´s results.
- And more...

We also offer a custom calculator service, for which we code and configure your calculator and send it to you ready to import. Find more information on our website www.moonsoft.es or send us an email to gestion@moonsoft.es

Enjoy Calc Builder

Moonsoft