Help ▾

Help

The Maintenance Mode is activated. Please don't forget to deactivate it as soon as you are done.

Here you'll find plenty of examples and documentation that should help you configure **cforms**.

This manual/help page is also available as a PDF document.

TABLE OF CONTENTS

1.Basic steps, a small guide

top

Admittedly, **cforms** is not the easiest form mailer plugin but it may be the most flexible. The below outline should help you get started with the default form.

1. First take a look at the default form
   - Verify that it contains all the fields you need, are they in the right order ⇅?
   - Check the field labels (field names), if needed make your adjustments
   - Check the flags for each field (check boxes to the right).
   - Want to include SPAM protection? Choose between Q&A, captcha add an input field accordingly and configure here.
2. Check if the email admin for your form is configured correctly.
3. Decide if you want the visitor to receive an auto confirmation message upon form submission.
4. Would you like to track form submission via the database?
5. Add the default form to a post or page.
6. Give it a whirl.

2.Inserting a form

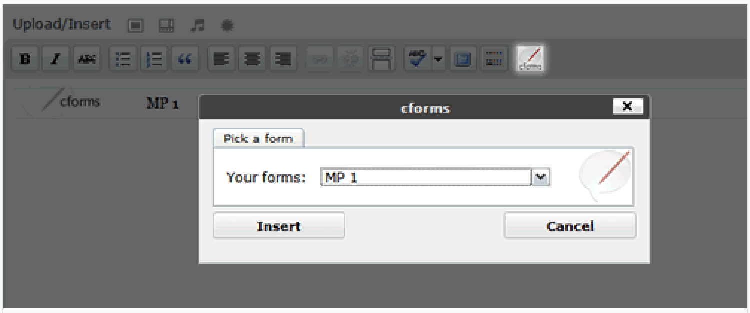
top

Editing posts and pages:

If you like to do it the 'code' way, make sure to use `<!--cforms name="XYZ"-->` to include them in your *Pages/Posts*. With `XYZ` being your form NAME.

A more elegant and safer way is to use the **TinyMCE Button** (double check if Button Support is enabled!).

For backwards compatibility, the old-fashioned way is still supported: `<!--cforms-->` for the first form and/or `<!--cformsX-->` for your other forms.



TinyMCE Support

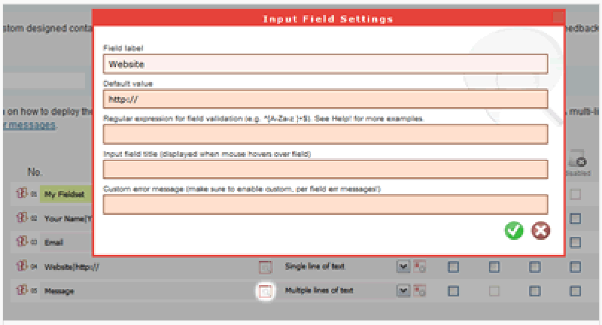
Via PHP function call:

Alternatively, you can specifically insert a form (into the sidebar for instance etc.) per the PHP function call `insert_cform('XYZ');` , or alternatively `insert_cform();` for the default/first form and/or `insert_cform('X');` for any other form.

Note: "**X**" represents the number of the form, starting with **2**, 3,4 ..and so forth.

3.Configuring form input fields

All supported input fields are listed below, highlighting the expected *formats* for their associated Field Names [*)]. Form labels (*Field Names*) permit the use of **HTML**, see examples below.


Input Field Settings

Field label: Website

Default value: http://

Regular expression for field validation (e.g. ^[A-Za-z]+$). See Help! for more examples.

Input field title (displayed when mouse hovers over field)

Custom error message (make sure to enable custom, per field err messages!)

A new *wizard like* mode allows you to configure more complex settings in case all the pipes "|" and pounds "#" are overwhelming.

form field

Text only elements (no input)

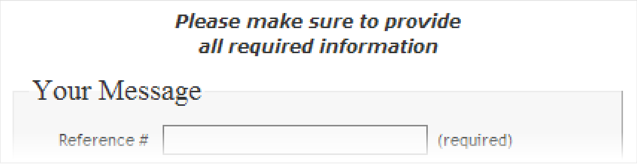
top

Format:	text paragraph | css class | optional style
Example:	`Please make sure...|mytextclass|font-size:9x; font-weight:bold;`
Example:	`Check <a href="http://mysite.com"> here </a> for more info. ||font-size:9x;`

HTML: the `text paragraph` supports HTML. If you need actual <, > in your text please use the proper HTML entity.

The above expression applies the custom class "`mytextclass`" **AND** the specific styles "`font-size:9x; font-weight:bold;`" to the paragraph.

If you specify a `css class`, you also need to define it in your current form theme file, here.


Please make sure to provide all required information

Your Message

Reference # (required)

form field

Javascript Date Picker

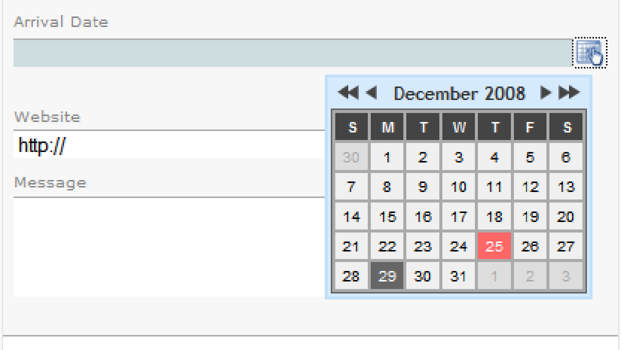
top

Format:	field name | default value | regular expression
Example:	`Arrival Date|mm/dd/yyyy|^[0-9][0-9]/[0-9][0-9]/[0-9][0-9][0-9][0-9]$`

The example above will set a *default value* of "mm/dd/yyyy" so users know the expected format. The **regexp** at the end ensures that only this format is accepted. **NOTE:** You also need to configure the date picker options to match the date format ("mm/dd/yyyy" !)

Supported Date Formats (see "Plugin Settings" tab)

Field	Full Form	Short Form
Year	yyyy (4 digits)	yy (2 digits)
Month	mmm (name)	mm (2 digits)
Day of Month	dd (2 digits)	


Arrival Date

Website: http://

Message

December 2008

S	M	T	W	T	F	S
30	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31	1	2	3

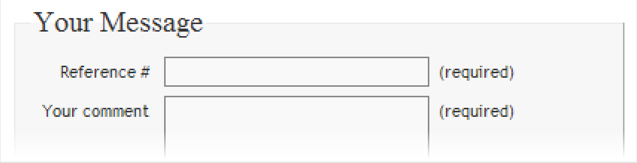
form field

Single, Password & Multi line input fields

top

Format:	field name | default value | regular expression
Example:	`Reference#|xxx-xx-xxx|^[0-9A-Z-]+$`
Example:	`Your <u>Full</u> Name||^[A-Za-z- \.]+$`
Example:	`<acronym title="We need your email address for confirmation.">Your EMail</acronym>`

You can of course omit the *default value* as in Example 2.


Your Message

Reference # (required)

Your comment (required)

form field

Select boxes & radio buttons

top

Format:	field name # option1 | value1 # option2 | value2 # option3...

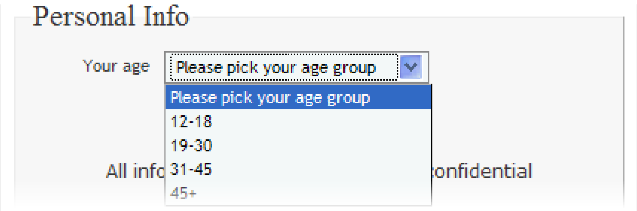
Example: `Your age#12-18|kiddo#19 to 30|young#31 to 45#45+ |older`

Omitting the `field name` will result in not showing a label to the left of the field.

The **option** parameter determines the text displayed to the visitor, **value** what is being sent in the email.

Is no **value** explicitly given, then the shown option text is the value sent in the email.

Select box marked "Required": Using a minus symbol `-` for the value (after **|**), will mark an option as invalid! Example:
`Your age#Please pick your age group|-#12 to 18|kiddo#19 to 30|young#31 to 45#45+ |older` .
"Please pick..." is shown but not considered a valid value.



Personal Info

Your age: Please pick your age group

- Please pick your age group
- 12-18
- 19-30
- 31-45
- 45+

All info onfidential

Multi select boxes

top

Format: field name # option1 | value1 # option2 | value2 # option3...

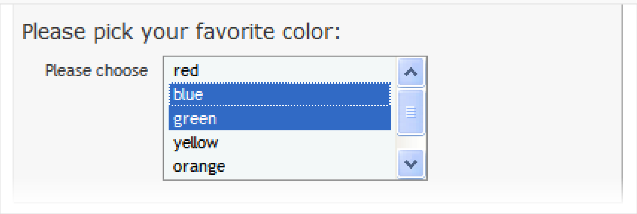
Example: `Pick#red#blue#green#yellow#orange`

Example: `<strong>Select</strong>#Today#Tomorrow#This Week#Next Month#Never`

Multi select fields can be set to **Required**. If so and unless at least one option is selected the form won't validate.

If `value1,2,..` are not specified, the values delivered in the email default to `option1,2,...` .

Examples for specific values could be the matching color codes: e.g. `red|#ff0000`



Please pick your favorite color:

Please choose: red, blue, green, yellow, orange

Check boxes

top

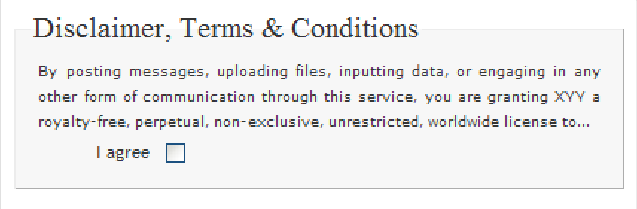
Format: field name left # field name right

Example: `#please check if you'd like more information`

You can freely choose on which side of the check box the label appears (e.g. `#label-right-only`).

If **both** left and right labels are provided, only the **right one** will be considered.

Check boxes can be flagged "**Required**" to support special use cases, e.g.: when you require the visitor to confirm that he/she has read term & conditions, before submitting the form.



Disclaimer, Terms & Conditions

By posting messages, uploading files, inputting data, or engaging in any other form of communication through this service, you are granting XYY a royalty-free, perpetual, non-exclusive, unrestricted, worldwide license to...

I agree □

Check box groups

top

Format: field name # chk box1 label|chk box1 value # chk box2 label ## chk box3...

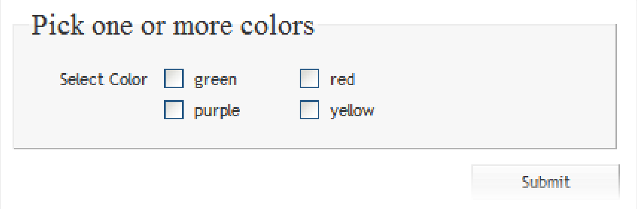
Example: `Select Color#green|00ff00 #red|ff0000 #purple|8726ac #yellow|fff90f`

Two # (`##`) in a row will force a new line! This helps to better structure your check box group.

Similar to **multi-select boxes** (see above), **Check box groups** allow you to deploy several check boxes (with their labels and corresponding values) that form one logical field. The result submitted via the form email is a single line including all checked options.

If no explicit **value** (text after the pipe symbol '|') is specified, the provided check box label is both label & submitted value.

None of the check boxes within a group can be made "Required".



Pick one or more colors

Select Color □ green □ red
□ purple □ yellow

CC: option for visitors

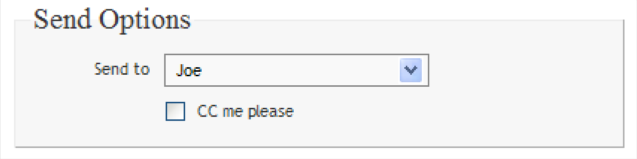
top

Format: field name left # field name right

Example: `#please cc: me`

If the visitor chooses to be CC'ed, **no** additional auto confirmation email (if configured) is sent out!

Please also see *check boxes* above.



Send Options

Send to: Joe

□ CC me please

Multiple form mail recipients

Note: This requires corresponding email addresses here!! top

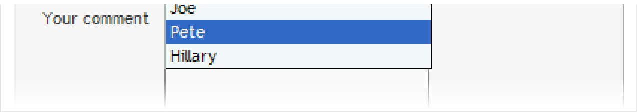
Format: field name # Name1 # Name2 # Name3...

Example: `Send to#Joe#Pete#Hillary`

The order of the names (1,2,3...) provided in the input field **directly** corresponds with the order of email addresses configured here.

Your message

Send to: Joe



Your comment
Joe
Pete
Hillary

form field

Hidden input fields

top

Format:	field name | default value
Example:	`extra-data|fixed,hidden text`
Example:	`post-data-meta|{custom_field_1}`

Hidden fields can contain fixed/preset values or **{variables}** which reference custom fields of posts or pages.

form field

Visitor verification (Q&A)

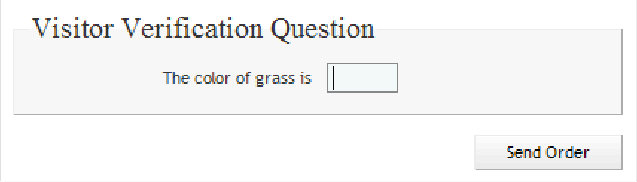
top

Format:	--
Example:	`--`

No `field name` required, the field has no configurable label per se, as it is determined at run-time from the list of **Question & Answers** provided here.

It makes sense to encapsulate this field inside a FIELDSET, to do that simply add a `New Fieldset` field before this one.

PLEASE NOTE that by default the captcha and visitor verification (Q&A) field are **not** shown for logged in users! This can be changed under global settings.



Visitor Verification Question

The color of grass is

form field

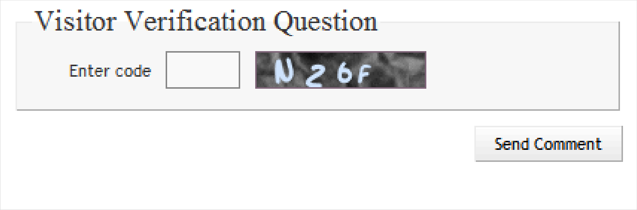
Captcha

top

Format:	field name
Example:	`Enter code`
Example:	`Spam Protection|title:We don't like spam bots|err:Please enter the CAPTCHA code correctly! If text is unreadable, try reloading.`

Alternatively or in addition to the above **Visitor verification** feature, you can have the visitor provide a captcha response.

PLEASE NOTE that by default the captcha and visitor verification (Q&A) field are **not** shown for logged in users! This can be changed under global settings.



Visitor Verification Question

Enter code

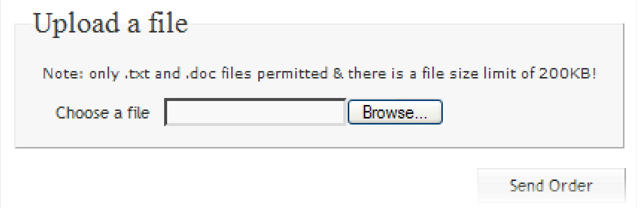
form field

Attachments / File Upload Box

top

Format:	form label
Example:	`Please select a file`

Please double-check the general settings for proper configuration of the `File Upload` functionality (allowed extensions, file size etc.).



Upload a file

Note: only .txt and .doc files permitted & there is a file size limit of 200KB!

Choose a file

form field

Tell a Friend input fields

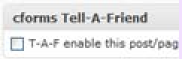
top

Format: of all 4 fields	field name | default value | regular expression

Fields:	`T-A-F * Your Name`
	`T-A-F * Your Email` *(make sure it's checked 'Email')*
	`T-A-F * Friend's Name`
	`T-A-F * Friend's Email` *(make sure it's checked 'Email')*

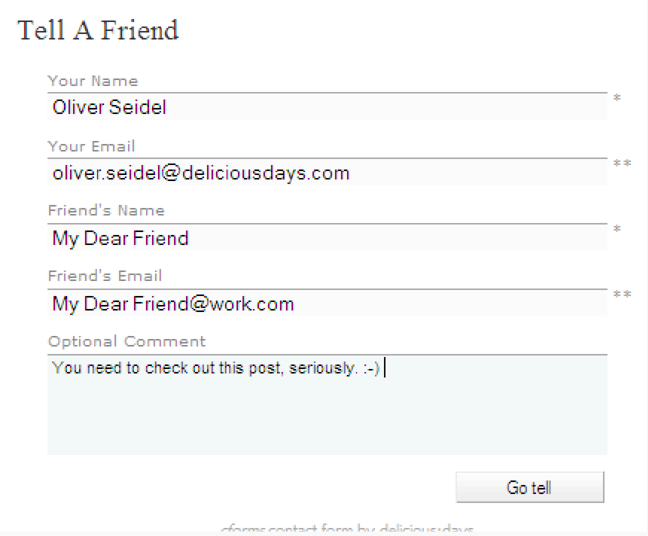
To get it working:

1. The Tell A Friend feature needs to be **enabled for the respective form** (*check if it's the right one!*), otherwise you won't see the above input fields in the [*Field Type*] select box.
2. The auto confirmation message will be used as a **message template** and needs to be defined. See example below.
3. There are three additional, *predefined variables* that can be used in the message template.
4. **Add the form** to your post/page php templates (see deployment options further below).
5. Tell-A-Friend **enable your posts/pages** by checking the T-A-F field in the WP post (page) editor.



cforms Tell-A-Friend

☐ T-A-F enable this post/pag



Tell A Friend

Your Name
Oliver Seidel *

Your Email
oliver.seidel@deliciousdays.com **

Friend's Name
My Dear Friend *

Friend's Email
My Dear Friend@work.com **

Optional Comment
You need to check out this post, seriously. :-)

Here is an example of how to setup the TXT part of the auto confirmation message as a Tell-A-friend template:

```
 Hello {Friend's Name},
{Your Name} left you this message:
{Optional Comment}
The message was sent in reference to {Title}:
{Excerpt}
{Permalink}
--
This email is sent, as a courtesy of website.com, located at http://website.com. The person who sent this email to you, {Your Name}, gave
an email address of {Your Email}. {Your Name} logged into website.com from IP {IP}, and sent the email at {Time}.
```

Note: In addition to the above TXT message you can, of course, add an HTML counterpart.

Recommended Implementation Options:

Alternative 1: The actual form will not show on the WP front page, but in the individual post/page view.

1. Simply add a `<?php insert_cform(#); ?>` (# = your form id) to your existing `single.php` and/or `page.php` template, e.g:

```
[...]
<?php the_content('<p>Read the rest of this entry »</p>'); ?>
<?php if ( is_tellafriend( $post->ID ) ) insert_cform(#); ?>
[...]
```

2. Suggestion: For a less crowded layout, optionally add some Javascript code to show/hide the form.

Alternative 2: A Tell-A-Friend link is shown under every eligible post/page, displayed both on the blog's front page & individual post & page view.

1. This requires a new WP page created (make note of the page ID or permalink), with its own page template (a clone of page.php will do). Add the following code to the new **page template**:

```
[...]
<?php the_content('<p>Read the rest of this page »</p>');?>
<h3> <?php echo 'E-Mail "'.get_the_title( $_GET['pid'] ).'" to a friend:'; ?> </p>
<?php if ( is_tellafriend( $_GET['pid'] ) ) insert_cform(#); ?>
[...]
```

2. In *single.php* & *index.php* and/or *page.php* add beneath the "the_content()" call the link to the new page created above, e.g.:

```
[...]
<?php the_content('<p>Read the rest of this entry »</p>'); ?>
<?php
if ( is_tellafriend( $post->ID ) )
    echo '<a href="[your-new-page]?&pid='.$post->ID.'" title="Tell-A-Friend form">Tell a friend!</a>';
?>
[...]
```

3. Replace **[your-new-page]** with **the permalink** of your newly created page.

WP Comment Feature input fields

form field

top

The beauty is, using one form, you can now offer your readers to either leave a comment behind or simply send a note to the post editor while being able to fully utilize all security aspects of cforms.

Special Fields:	**`Comment Author`**
	`Author's Email`
	`Author's URL`
	`Author's Comment`
	`Subscribe To Comments` `(if plugin installed)`
	`Comment Luv` `(if plugin installed)`

Format: for top 4 fields	field name | default value | regular expression
Special Field:	**`Select: Email/Comment`**
Format:	field name # text *comment* | 0 # text *to author* | 1
Example:	`Send as#regular comment|0#email to post author|1`

To get it working:

1. Turn on the WP Comment feature for the given form. (*Make sure it's the right one!*), otherwise you won't see the above input fields in the [*Field Type*] select box.

2. Modify this form to include all the necessary (new) input fields, make them required or not, add regexp, anti SPAM fields or even custom err messages. All up to you. Or better yet, start with the built-in preset: "**Advanced: WP comment...**" form.

3. Edit your WP Theme template for comments. Remove the current **form tag** entirely (`<form action="...</form>`). Instead replace with a PHP call to cforms: `<?php insert_cform(X); ?>` with **X** being omitted if the form is your default form or starting at **'2'** (with single quotes!) for any subsequent form #. See example comments.php here!

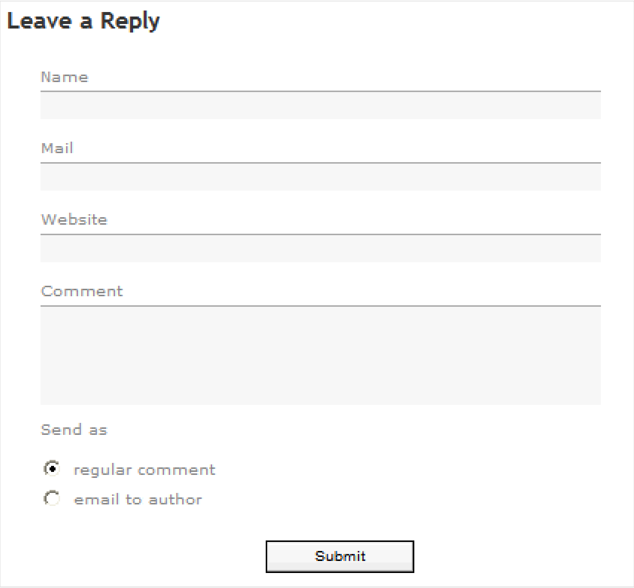
4. Double check the extended WP comment feature settings here (especially the Ajax specific ones!).

5. **Important:** To make Ajax work in case there are no comments yet, make sure that the comment container **is always** being rendered.

Suggestions:

1. I recommend you choose the **wide_form.css** theme under the Styling menu. And adjust to your liking.

2. If you intend to make certain fields "required", I further recommend you add the text "*required*" to the input field label and set this style: `span.reqtxt, span.emailreqtxt`



Leave a Reply

Name

Mail

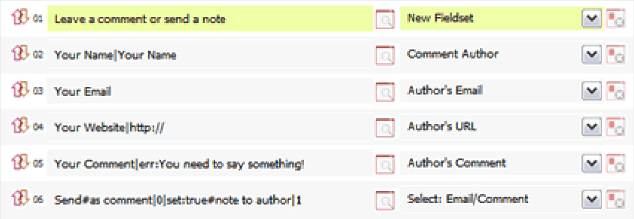
Website

Comment

Send as

● regular comment

○ email to author



01 Leave a comment or send a note — New Fieldset

02 Your Name|Your Name — Comment Author

03 Your Email — Author's Email

04 Your Website|http:// — Author's URL

05 Your Comment|err:You need to say something! — Author's Comment

06 Send#as comment|0|set:true#note to author|1 — Select: Email/Comment

Example Configuration

`{...` to `display:none;` (using the CSS editor on the *Styling* page)

Other comment plugins? cforms' WP comment feature supports the following comment plugins: Comment Luv, Subscribe To Comment & WP Ajax Edit Comments.

Tutorial: Here you'll find a comprehensive write up by Erum Munir on how to use cforms in combination with the Subscribe-To plugin, and a more general one for using cforms as a stand in replacement for the default WP comment functionality.

form field Fieldsets

top

Fieldsets are definitely part of good form design, they are form elements that are used to create individual sections of content within a given form.

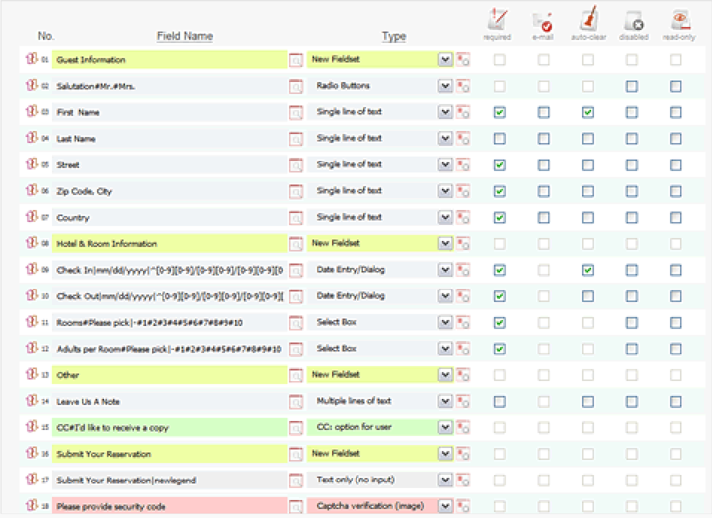
Format:	fieldset name
Example:	`My Fieldset`

Fieldsets can begin anywhere, simply add a **New Fieldset** field between or before your form elements.

Fieldsets do not need to explicitly be closed, a **New Fieldset** element will automatically close the existing (if there is one to close) and reopen a new one.

End Fieldset can be used, but it works without just as well.

If there is no closing **End Fieldset** element, the plugin assumes that it needs to close the set just before the submit button



<table>
<tr><th>No.</th><th>Field Name</th><th>Type</th><th>required</th><th>e-mail</th><th>auto-clear</th><th>disabled</th><th>read-only</th></tr>
<tr><td>01</td><td>Guest Information</td><td>New Fieldset</td><td>□</td><td>□</td><td>□</td><td>□</td><td>□</td></tr>
<tr><td>02</td><td>Salutation#Mr.#Mrs.</td><td>Radio Buttons</td><td>□</td><td>□</td><td>□</td><td>□</td><td>□</td></tr>
<tr><td>03</td><td>First Name</td><td>Single line of text</td><td>☑</td><td>□</td><td>☑</td><td>□</td><td>□</td></tr>
<tr><td>04</td><td>Last Name</td><td>Single line of text</td><td>□</td><td>□</td><td>□</td><td>□</td><td>□</td></tr>
<tr><td>05</td><td>Street</td><td>Single line of text</td><td>☑</td><td>□</td><td>□</td><td>□</td><td>□</td></tr>
<tr><td>06</td><td>Zip Code, City</td><td>Single line of text</td><td>☑</td><td>□</td><td>□</td><td>□</td><td>□</td></tr>
<tr><td>07</td><td>Country</td><td>Single line of text</td><td>☑</td><td>□</td><td>□</td><td>□</td><td>□</td></tr>
<tr><td>08</td><td>Hotel &amp; Room Information</td><td>New Fieldset</td><td>□</td><td>□</td><td>□</td><td>□</td><td>□</td></tr>
<tr><td>09</td><td>Check In|mm/dd/yyyy|^[0-9][0-9]/[0-9][0-9]/[0-9][0-9][0</td><td>Date Entry/Dialog</td><td>☑</td><td>□</td><td>☑</td><td>□</td><td>□</td></tr>
<tr><td>10</td><td>Check Out|mm/dd/yyyy|^[0-9][0-9]/[0-9][0-9]/[0-9][0-9][</td><td>Date Entry/Dialog</td><td>☑</td><td>□</td><td>□</td><td>□</td><td>□</td></tr>
<tr><td>11</td><td>Rooms#Please pick|-#1#2#3#4#5#6#7#8#9#10</td><td>Select Box</td><td>☑</td><td>□</td><td>□</td><td>□</td><td>□</td></tr>
<tr><td>12</td><td>Adults per Room#Please pick|-#1#2#3#4#5#6#7#8#9#10</td><td>Select Box</td><td>☑</td><td>□</td><td>□</td><td>□</td><td>□</td></tr>
<tr><td>13</td><td>Other</td><td>New Fieldset</td><td>□</td><td>□</td><td>□</td><td>□</td><td>□</td></tr>
<tr><td>14</td><td>Leave Us A Note</td><td>Multiple lines of text</td><td>□</td><td>□</td><td>□</td><td>□</td><td>□</td></tr>
<tr><td>15</td><td>CC#I'd like to receive a copy</td><td>CC: option for user</td><td>□</td><td>□</td><td>□</td><td>□</td><td>□</td></tr>
<tr><td>16</td><td>Submit Your Reservation</td><td>New Fieldset</td><td>□</td><td>□</td><td>□</td><td>□</td><td>□</td></tr>
<tr><td>17</td><td>Submit Your Reservation|newlegend</td><td>Text only (no input)</td><td>□</td><td>□</td><td>□</td><td>□</td><td>□</td></tr>
<tr><td>18</td><td>Please provide security code</td><td>Captcha verification (image)</td><td>□</td><td>□</td><td>□</td><td>□</td><td>□</td></tr>
</table>

form field Using regular expressions with form fields

top

A regular expression (regex or regexp for short) is a special text string for describing a search pattern, according to certain syntax rules. Many programming languages support regular expressions for string manipulation, you can use them here to validate user input. Single/Multi line input fields:

<table>
<tr><td><b>Format:</b></td><td>field name | default value | regular expression</td></tr>
<tr><td><b>Example:</b>
<b>US zip code</b></td><td><code>zip code||^\d{5}$)|(^\d{5}-\d{4}$</code></td></tr>
<tr><td><b>Example:</b>
<b>US phone #</b></td><td><code>phone||^[\(]?(\d{0,3})[\)]?[\s]?[\-]?(\d{3})[\s]?[\-]?(\d{4})[\s]?[x]?(\d*)$</code></td></tr>
<tr><td><b>Special Example:</b>
<b>comparing two input fields</b></td><td><code>please repeat email||cf2_field_2</code></td></tr>
</table>

Important:
If you need to compare two input fields (e.g. email verification): simply use the regexp field (see special example above, to point to the HTML element ID of the field you want to compare the current one to. To find the HTML element ID you would have to look into the html source code of the form (e.g. `cf2_field_2`).

GENERAL:

Ensure that the input field in question is tagged '**Required**'!

`^` and `$` define the start and the end of the input

" `ab*` ": ...matches a string that has an "a" followed by zero or more "b's" ("a", "ab", "abbb", etc.);

" `ab+` ": ...same, but there's at least one b ("ab", "abbb", etc.);

" `[a-d]` ": ...a string that has lowercase letters "a" through "d"

More information can be found here, a great regexp repository here.

4.Custom error messages & input field titles

top

On top of their labels, input fields can have titles, too. Simply append a `|title:XXX` to a given field configuration string.

If you like to add custom error messages (next to your generic success and error messages) for your input fields, simply append a `|err:XXX` to a given *definition string/field name*. HTML is supported.

Please note the order of these special attributes, first `|title:XXX` (if applicable), then `|err:XXX` .

<table>
<tr><td><b>Format:</b></td><td>field name <b>|title: <i>your title here</i> |err: <i>your error message</i></b></td></tr>
<tr><td><b>Example 1:</b></td><td><code>Your Name|title:Only alphabetic characters allowed!</code></td></tr>
<tr><td><b>Example 2:</b></td><td><code>Your Name|title:Please provide your first and last name!|err:Please enter your full name.</code></td></tr>
<tr><td><b>Example 3:</b></td><td><code>Your age#12-18|kiddo#19 to 30|young#31 to 45#45+ |older|err: your age is <strong>important</strong> to us.</code></td></tr>
</table>

Note:

Custom error messages can be applied to any input field that can be flagged "**Required**", **titles** to any input field.

5.Advanced: cforms APIs & (Post-)Processing of submitted data

top

cforms currently supports one API call to retrieve stored form data from the cforms database tables.

API Function	
`get_cforms_entries( $fname, $from, $to, $sort, $limit )`	This function will return a set of stored form submissions in a multi-dimensional array.
`$fname`	text string (regexp pattern) :: e.g. the form name
`$from, $to`	DATETIME string (format: Y-m-d H:i:s) :: date & time defining the target period, e.g. **2008-09-17 15:00:00**
`$sort`	text :: **form_id**, **id**, **sub_date**, **ip** or **email**
`$limit`	number :: limits the number of results, '' (empty) = no limits!
Examples	
`$array = get_cforms_entries('','','','','') /* all data, no filters */`	
`$array = get_cforms_entries('contact','','','sub_date',5) /* last 5 submissions of "my contact form", order by date */`	
`$array = get_cforms_entries('',date ("Y-m-d H:i:s", time()-(3600*2)),'','','') /* all submissions in the last 2 hours */`	

(Post-)Processing of submitted data is really for hard core deployments, where *real-time manipulation* of a form & fields are required. If you require the submitted data to be manipulated, and or sent to a 3rd party or would like to make use of the data otherwise, here is how:

Available Functions (see **my-functions.php** file (plugin root directory), including examples)	
`my_cforms_filter()`	function gets triggered **after** user input validation and cforms processing
`my_cforms_ajax_filter()`	function gets called **after** input validation, but **before** further processing (nonAjax)
`my_cforms_action()`	function gets called **after** input validation, but **before** further processing (Ajax)
`my_cforms_logic()`	function gets called **at** various stages of input processing

my-functions.php can reside in your /plugins/custom-cforms folder to protect it from future (auto) upgrades.

6.Advanced: Real-time creation of dynamic forms

top

Again, this is for the advanced user who requires ad-hoc creation of forms.

A few things to note on dynamic forms:

1. Dynamic forms only work in **non-Ajax** mode.
2. Each dynamic form references and thus requires **a base form defined** in the cforms form settings. All its settings will be used, except the form (&field) definition.
3. Any of the form fields described in the plugins' **HELP!** section can be dynamically generated.
4. Function call to generate dynamic forms: `insert_custom_cform($fields:array,$form-no:int);` with

`$form-no` : empty string for the first (default) form and **2**,3,4... for any subsequent form
`$fields` :

```
$fields['label'][n]      = 'field name';           field name format described above
$fields['type'][n]       = 'input field type';     default: 'textfield';
$fields['isreq'][n]      = true|false;             default: false;
$fields['isemail'][n]    = true|false;             default: false;
$fields['isclear'][n]    = true|false;             default: false;
$fields['isdisabled'][n] = true|false;             default: false;
$fields['isreadonly'][n] = true|false;             default: false;

n = 0,1,2...
```

Form input field types ('type'):

Basic fields		**Special T-A-F fields**	
Text paragraph:	`textonly`	T-A-F * Your Name:	`yourname`
Single input field:	`textfield`	T-A-F * Your Email:	`youremail`
Multi line field:	`textarea`	T-A-F * Friend's Name:	`friendsname`
Hidden field:	`hidden`	T-A-F * Friend's Name:	`friendsemail`
Password field:	`pwfield`		
Date picker field:	`datepicker`	**WP Comment Feature**	
Check boxes:	`checkbox`	Comment Author:	`author`
Check boxes groups:	`checkboxgroup`	Author's Email:	`email`
Drop down fields:	`selectbox`	Author's URL:	`url`
Multi select boxes:	`multiselectbox`	Author's Comment:	`comment`
Radio buttons:	`radiobuttons`	Select: Email/Comment:	`send2author`
'CC' check box *):	`ccbox`	Subscribe To Comments:	`subscribe`
Multi-recipients field *):	`emailtobox`	Comment Luv:	`luv`
Spam/Q&A verification *):	`verification`		
Spam/captcha verification *):	`captcha`		

File upload fields *):	`upload`
Begin of a fieldset:	`fieldsetstart`
End of a fieldset:	`fieldsetend`

*) *Should only be used **once** per generated form!*

Simple example:

```
$fields = array();

$formdata = array(
                array('Your Name|Your Name','textfield',0,1,0,1,0),
                array('Your Email','textfield',0,0,1,0,0),
                array('Your Message','textarea',0,0,0,0,0)
                );

$i=0;
foreach ( $formdata as $field ) {
        $fields['label'][$i]        = $field[0];
        $fields['type'][$i]         = $field[1];
        $fields['isdisabled'][$i]   = $field[2];
        $fields['isreq'][$i]        = $field[3];
        $fields['isemail'][$i]      = $field[4];
        $fields['isclear'][$i]      = $field[5];
        $fields['isreadonly'][$i++] = $field[6];
}

insert_custom_cform($fields,'');    //Call default form with two defined fields
```

More advanced example (file access)**:**

```
$fields['label'][0]  ='Your Name|Your Name';
$fields['type'][0]   ='textfield';
$fields['isreq'][0]  ='1';
$fields['isemail'][0]='0';
$fields['isclear'][0]='1';
$fields['label'][1]  ='Email';
$fields['type'][1]   ='textfield';
$fields['isreq'][1]  ='0';
$fields['isemail'][1]='1';
$fields['label'][2]  ='Please pick a month for delivery:||font-size:14px; padding-top:12px; text-align:left;';
$fields['type'][2]   ='textonly';

$fields['label'][3]='Deliver on#Please pick a month|-#';

$fp = fopen(dirname(__FILE__).'/months.txt', "r"); // Need to put this file into your themes dir!
while ($nextitem = fgets($fp, 512))
        $fields['label'][3] .= $nextitem.'#';

fclose ($fp);

$fields['label'][3]  = substr( $fields['label'][3], 0, strlen($fields['label'][3])-1 );  //Remove the last '#'$fields['type'][3]   ='selectbox';
$fields['isreq'][3]  ='1';
$fields['isemail'][3]='0';

insert_custom_cform($fields,5);    //Call form #5 with new fields
```

With `month.txt` containing all 12 months of a year:

```
January
February
March
...
```

7.Using variables in email subject and messages

top

Subjects and messages for emails both to the form admin as well as to the visitor (auto confirmation, CC:) support insertion of pre-defined variables and/or any of the form input fields.

Note that the variable names are case sensitive!

Predefined variables:	
{BLOGNAME}	Inserts the Blog's name.
{Form Name}	Inserts the form name (per your configuration).
{PostID}	Inserts the ID of the post the form is shown in.
{Referer}	Inserts the HTTP referer information (if available).
{Page}	Inserts the WP page the form was submitted from.
{Date}	Inserts the date of form submission (per your general WP settings).
{Time}	Inserts the time of form submission (per your general WP settings).
{IP}	Inserts visitor IP address.

`{ID}`	Inserts a unique and referenceable form ID (provided that DB Tracking is enabled!)
`{CurUserID}`	Inserts the ID of the currently logged-in user.
`{CurUserName}`	Inserts the Name of the currently logged-in user.
`{CurUserEmail}`	Inserts the Email Address of the currently logged-in user.
Special:	A single " `.` " (period) on a line inserts a blank line.
Predefined variables for Tell-A-Friend forms:	
`{Permalink}`	Inserts the URL of the WP post/page.
`{Author}`	Inserts the Author's name (*Nickname*).
`{Title}`	Inserts the WP post or page title.
`{Excerpt}`	Inserts the WP post or page excerpt.
Custom variables (referencing input fields):	
Alternatively to the cforms predefined variables, you can also reference data of any of your form's input fields by one of the 3 ways described below.	
`{field label}`	With *field label* being the exact field label as it is being tracked and sent in the admin email!
`{XYZ}`	In case you're using the custom input field NAMES & ID's, the reference is the id: of the field.
`{_fieldNN}`	With *NN* being the position of the field on the form configuration page.

Example:
Suppose this is the input field definition string: `Your Website[id:homepage]|http://`
The corresponding variables would be: `{Your Website}` , `{homepage}` , or `{_field4}` (assuming it is on the 4th position) respectively.

Important: If you are using multiple input fields with **the same** recorded field label (you can always check the "Tracking" menu tab for how the fields are stored), e.g:

Size#250gr.#500gr#1kg circa
Size#450gr.#700gr#1.2kg circa
Size#650gr.#800gr#1.5kg circa

Results in the first field labeled '**Size**' to be addressed with `{Size}` . The second instance of '**Size**' can be addressed by `{Size__2}` , and so on...

Here is an example for a simple Admin HTML message *(you can copy and paste the below code or change to your liking)*:

HTML code:

<p style="background:#fafafa; text-align:center; font:10px arial">a form has been submitted on {Date}, via: {Page} [IP {IP}]</p>

Output:

a form has been submitted on June 13, 2007 @ 9:38 pm, via: / [IP 184.153.91.231]

Note: With this limited message you'd want to enable the option "Include pre formatted form data table in HTML part"

Here is another example for a more detailed Admin HTML message:

HTML code:

<p>{Your Name} just submitted {Form Name}. You can get in touch with him/her via <a href="mailto:{Email}">{Email}</a> and might want to check out his/her web page at <a href="{Website}">{Website}</a></p>

<p>The message is:<br/ >
{Message}</p>

Output:

John Doe just submitted MY NEW FORM. You can get in touch with him/her via john.doe@doe.com and might want to check out his/her web page at http://website.com

The message is:
Hey there! Just wanted to get in touch. Give me a ring at 555-...

Note: With this more detailed message you can disable the option "Include pre formatted form data table in HTML part" since you already have all fields covered in the actual message/header.

And a final example for a HTML auto confirmation message:

HTML code:

<div style="text-align:center; color:#aaa; border-bottom:1px solid #aaa"> <strong>auto confirmation message, {Date}</strong>
</div><br />

<p><strong>Dear {Your Name},</strong></p>
<p>Thank you for your note!</p>
<p>We will get back to you as soon as possible.</p>

Output:

auto confirmation message, June 13, 2007 @ 5:03 pm

Dear John Doe,

Thank you for your note!

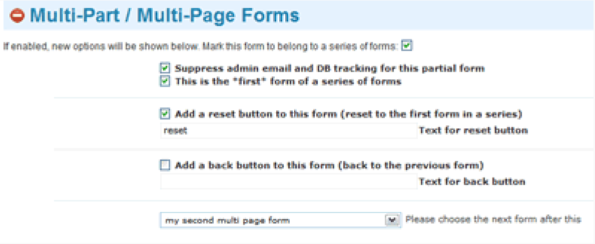
We will get back to you as soon as possible.

8.Multi page forms

top

Multi-page-forms support chaining of several forms and gather user input across all linked forms. Inserting a multi page form is easy, simply insert the **first form** of the series into your post or page.

Multi-part/-page form features:
Defining first, next and last form via configuration
Overriding "next form" at run-time (dynamically) via my-functions.php
Optionally send/suppress partial admin emails on a per form basis
A form reset button
A form back button



Multi-Part / Multi-Page Forms

If enabled, new options will be shown below. Mark this form to belong to a series of forms: ☑

☑ Suppress admin email and DB tracking for this partial form
☑ This is the "first" form of a series of forms

☑ Add a reset button to this form (reset to the first form in a series)
reset — Text for reset button

□ Add a back button to this form (back to the previous form)
Text for back button

my second multi page form — Please choose the next form after this

Example (eg. using 3 forms):	
`form 1,2,3:`	select main check box to enable as multi-part forms
`form 1:`	(a) check "Suppress admin email.." (b) check "This is the first form.." (c) select "form 2" as next form
`form 2:`	(a) check "Suppress admin email.." (b) select "form 3" as next form
`form 3:`	(a) make sure to not! have "Suppress admin email.." selected (b) select "last form" to stop further routing
Optionally add Reset & Back buttons where appropriate.	
Further, it makes sense to change "Submit Button" text (to e.g. "Continue") & the "success message" to rather announce the next form.	

Important Notes:

Accessing **{custom variables}** in the final form differs from how you would reference these in individual forms. Use the **mail()** example in my-functions.php to examine the user data array; e.g. **{Email}** would become **{cf_form_Email}** (for the first form of the series).

File attachments will not be included in the admin email unless the upload fields are on the last form. However, they will be stored and tracked.

Once the multi page form support is enabled, **Ajax is being disabled** for this form.

9.Styling Your Forms (CSS theme files)

top

Please see the Styling page for theme selection and editing options.

cforms comes with a few theme examples (some of them may require adjustments to work with **your** forms!) but you can of course create your own theme file -based on the default **cforms.css** file- and put it in the `/styling` directory.

With v8.5+ cforms supports a separate custom user folder to store your tailored CSS, font and image files! Simply create the folder: **/plugins/cforms-custom** and move your CSS (including **all** images!), font & background image files (CAPTCHA) to it.

You might also want to study the PDF guide on cforms CSS & a web screencast I put together to give you a head start.

Your form **doesn't** look like the preview image, or your individual changes don't take effect, check your global WP theme CSS! It may overwrite some or many cforms CSS declarations. If you don't know how to trouble shoot, take a look at the Firefox extension "Firebug" - an excellent CSS troubleshooting tool!

10.Need more help?

top

For up-to-date information first check the FAQs & cforms forum and comment section on the plugin homepage.

*For more information and support, visit the **cforms** support forum . Translation provided by Oliver Seidel, for updates check here.*

Version v10.1