

Using Tectite Formail

Script
Configuration
Guide



Introduction

The Tectite Formmail Script is a very comprehensive PHP script that will run on all web hosting servers that have PHP installed.

Unlike script generators, such as Forms To Go or Coffee-Cup, the script only has to be configured once and can then be used to process ANY and ALL forms you may create on your WebPlus site. In this respect, it works in much the same way as Serif's own Web Resources form script (SWR) but without the limitations, or possible "off-line" status of SWR. Script generators often require you to create an associated script for each form in your website. This can get a little confusing if you have a number of different forms, or if you wish to change your forms in the future.

How the Script Works.

The script is ready-made and only requires you to edit three parameters to make it function correctly. The script itself sits on your web server and receives input from any form you direct to the script. The forms you create in WebPlus can contain hidden fields in which you can send additional instructions to the script to process your data in different ways. This is a very flexible solution for most people because it doesn't matter how many form fields you have, the script will always handle them correctly. Once data has been sent to the script, it will send that information to any number of email addresses you specify. Furthermore, you can choose to have the information sent to your recipients as formatted HTML pages or as plain text. These options can be set differently for every form in your website. So, if you want different forms handled in different ways, all you have to do is add some extra hidden fields to your WebPlus forms.

Other features of the script will allow you to store form data in a database or a CSV file for later retrieval or further processing. Your forms can also handle file uploads. In most cases, any uploaded files will be sent to recipients along with the other form data. However, you can if you wish choose to save uploaded files to a directory on your server.

Where to get the script

You can download the script from [the Tectite website](#). You can, if you wish, pay a small fee and have the script configured for you. However, the edits you need to make are so simple, you may as well do it yourself. You should also check with your web host to see if they have the script available as a "quick install" application. If they do, it will usually be configured for basic use on your domain automatically. If your host gives you this option, the only time you will have to edit the script is if you choose to implement some of the other options that the script offers.

If you download the script it will be a PHP file. This file can be opened with any plain text editor but my recommendation would be to use an editor such as [notepad++](#) if you are a Windows user, or [text wrangler](#) if you are a Mac user. Click the links to download these recommended editors.

Basic Configuration

When you download the script it will initially be in the form of a zip file. Save this file to a convenient location (*the desktop is good*) and double-click to decompress the file. This will create a folder in which you will find the main script file (***formmail.php***) and a sample form.

To open the script, right-click on the ***formmail.php*** file and select “***open with***”. From the context menu choose your text editor application.

Don’t worry if the script looks a little daunting – it really is easy to edit and you don’t need to have any understanding of the code. Scroll down the script until you see a section headed “***Configuration***”. It will look something like this:

```
/* *****  
/* CONFIGURATION (do not alter this line in any way!!!)          */  
/* *****  
* This is the *only* place where you need to modify things to use formmail.php  
* on your particular system. This section finishes at "END OF CONFIGURATION".  
* Help for all settings can be found on our website:  
* http://www.tectite.com/fmdoc/index.php  
*  
* Also, above each setting is a direct URL to the help information for the  
* setting.  
/* *****
```

Immediately under this heading you will see the lines of code that can be edited. The first item you should edit is the ***\$TARGET_EMAIL*** parameter. It looks like this:

```
$TARGET_EMAIL = array(EMAIL_NAME."@yourhost\.com$");
```

This is an important instruction as it tells the script that it may ***only*** send form results to email addresses that match this parameter. In the above example, the script is being told that it may ***ONLY*** send form data to email addresses that end with “***@yourhost.com***”. So, you will change this instruction so the script will only send form results to ***your*** email domain. For example: If your normal email address is fred.bloggs@bloggsdomain.com you would change the parameter as follows:

```
$TARGET_EMAIL = array(EMAIL_NAME."@bloggsdomain\.com$");
```

The only thing you must remember is to put a “\” BEFORE the “.” And put a “\$” after “com”

What does the TARGET_EMAIL parameter do?

The TARGET_EMAIL parameter is an important security feature designed to prevent spammers from using your script as a “spam gateway”. What it does is ensure that any forms sent to the script can only be sent to email addresses that MATCH this parameter. So, if a form was directed at the script with a recipient set to **george@hotmail.com** the form data would not be sent because **hotmail** does not match the parameter of “**bloggsdomain**”

To explain this a little further, when you create a form in WebPlus, you would create a **hidden field** named **recipients**. The value of that field will be the **full** email address that you want your form data sent to. When the script receives the form data, it will look at the **recipients hidden field** and compare the email address it finds with the **TARGET_EMAIL parameter**. If the two match, then the form results will be sent. If they don’t match, the email will not be sent and an error report will be sent to the site administrator.

So how can a spammer change the recipients hidden field?

Simple. When someone views your form in a web browser, they can use the browser’s “view page source” function. This will reveal the form fields, including the hidden fields. They will also see the URL of your form script. So, all a spammer has to do is create an HTML form and substitute the value of your hidden field to the email address of someone he want’s to spam. He can then submit the form to your script and, hey-presto, your script will forward the spammer’s unsolicited message to his target.

If you set up the TARGET_EMAIL parameter, spammers will not be able to use this trick because your script will reject all emails that do not match the parameter you’ve set within the script.

A Word of Warning!

If you have a web-based email account such as hotmail, gmail or yahoo, DO NOT SET THE TARGET_EMAIL PARAMETER to direct form results to the DOMAIN ONLY. Again, this is because spammers could discover that your script will send form results to ANY email address that ends in hotmail, gmail or yahoo. This would allow them to spam literally millions of people worldwide. Instead, use an EXACT MATCH parameter for your TARGET_EMAIL. Here is an example if you wanted your form results to go to george@hotmail.com

```
$TARGET_EMAIL = array("^george@hotmail\.com$");
```

Notice the syntax: we’ve replaced the array with (“^george@hotmail \.com\$”) It’s important to use the opening and closing quotes as well as placing the caret symbol before the email address. Now, any forms submitted to the script can only be sent to that specific email address.

The DEF_ALERT parameter

The next thing you need to configure is the DEF_ALERT parameter. You will find it immediately below the TARGET_EMAIL line in the script. This is what it looks like:

```
define("DEF_ALERT","");
```

This is where you tell the script where to send any alert or error messages that the script may generate. Typical of such messages may be during testing stage if one of your form parameters is incorrect, or in general use, if someone tries to spam your script. Simply enter the full email address of your site administrator in between the quotes. Here is an example:

```
define("DEF_ALERT","admin@yourdomain.com");
```

The AT_MANGLE parameter

This is an OPTIONAL parameter, but one that I recommend you set. Again, it's a security measure designed to defeat spam-bots – those pesky robots that visit your site looking for email addresses. Remember I said that spammers could look at your page source to see where your form results are being sent? Well, spam-bots do the same thing but are automated. They work by scanning the source code of your web pages to see if anything resembling an email address can be found. Clearly, if such a bot scans your form and sees an email address in your **recipients hidden field**, it can harvest that address and add it to its hit-list of email addresses. The AT_MANGLE parameter helps defeat this kind of attack by disguising the "@" symbol in the email address. The "@" symbol is usually the giveaway to spam-bots that an email address exists on your page. By removing the symbol your email address will most likely not be recognised.

The AT_MANGLE parameter is used to tell the script to convert any pre-defined string of characters or symbols that it finds in a **recipients hidden form field** into an "@" symbol. So, if you define an AT_MANGLE as [^23h5* you would use this string of characters in place of the "@" symbol in any email address you enter in your form's **recipients hidden field**. This is how the AT_MANGLE parameter looks in the script:

```
define("AT_MANGLE","");
```

Simply enter your defined string of characters or symbols between the last pair of quotes, like this:

```
define("AT_MANGLE","[^23h5*");
```

When it comes to creating your form you would enter email addresses in the recipients field as
emailname[^23h5*yourdomain.com

Completion of Configuration

Once you've made these few edits to your script, you should save it. A good practice is to rename the script to something less obvious. Instead of **formmail.php** you could name it something like **emailforward.php**. In fact, you can give the script any name you like as long as it ends with .php.

At this point, you could upload the script to your server just to test that it works. Use any FTP program, such as [Filezilla](#), to do this, or you could use your host's file manager application to upload.

Once uploaded you can test that the script can send you emails by typing the following URL into your web browser:

<http://yourdomain.com/formmail.php?testalert=1>

Replace **yourdomain.com** with the actual domain to where you uploaded the script, and change **formmail.php** to the name you gave your script if you changed it. The above assumes you placed the script in the root of your domain. If you uploaded it to a scripts folder, add the full path to the script:

<http://yourdomain.com/scriptfolder/formmail.php?testalert=1>

If all is well with your script, you should receive a message in your inbox (*or maybe your spam folder*) that looks like this:

To: [your DEF_ALERT email address](#)
From: formmail@yourdomain.com

This is a test alert message
Loaded language is English (builtin)
PHP version is 5.2.17
FormMail version is 8.06
Server type: non-Windows

DOCUMENT_ROOT: [the full path to your domain](#)
SCRIPT_FILENAME: [the full path and the name of your script](#)
PATH_TRANSLATED: -not set-
REAL_DOCUMENT_ROOT: [the full path to your domain](#)

User IP address was: Your IP Address

This is your confirmation that your script is set up and ready to send you form results. In the next section we take a look at how to set up forms in WebPlus and direct those forms to your script.

CREATING FORMS IN WEBPLUS TO USE YOUR SCRIPT

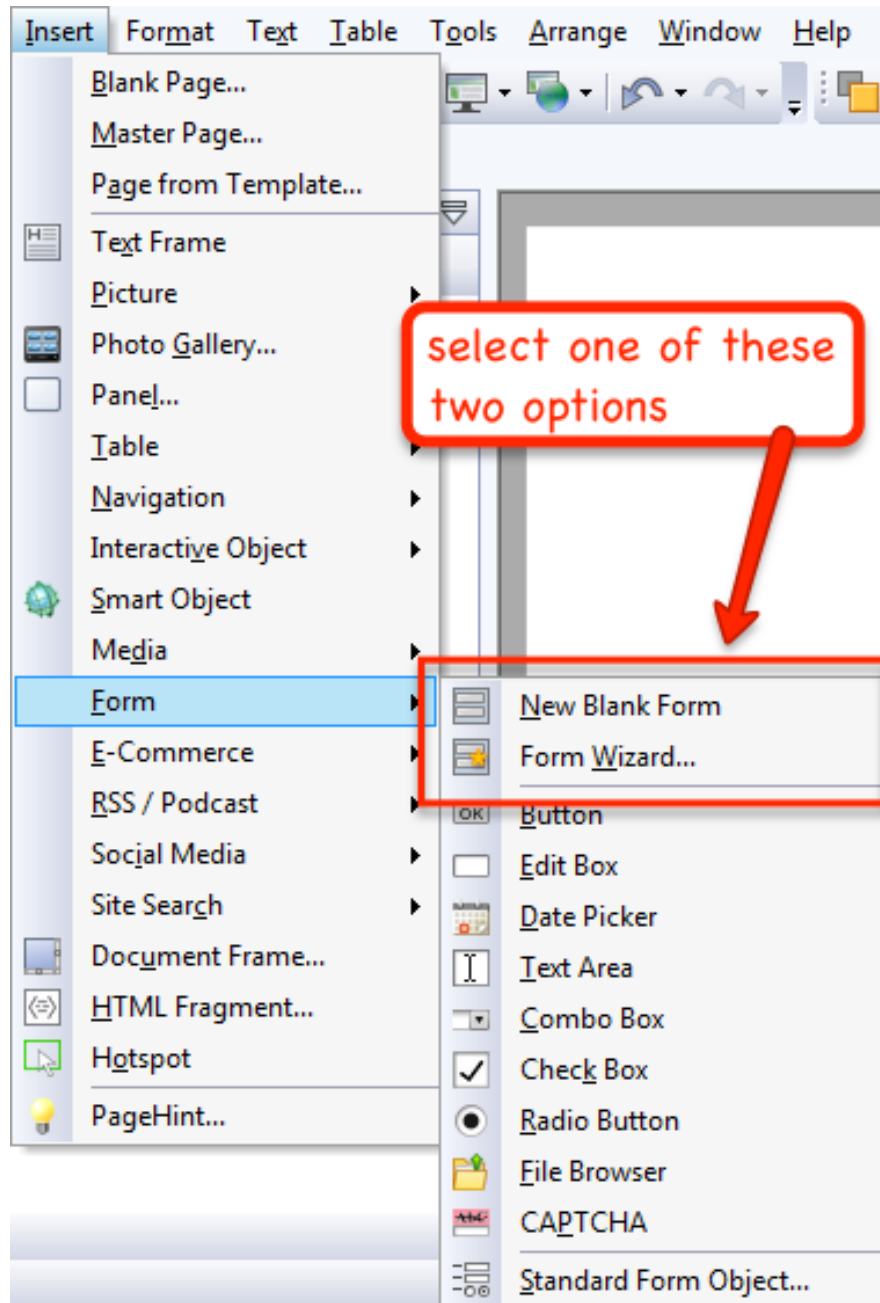
Tip

If you haven't already done so, I recommend that you upload your script to your server and test that you are able to receive alert emails. By doing this you will be sure that any alerts can be processed by the script and forwarded to you without problem.

It also allows you to test your forms from within WebPlus during form creation. If you don't upload your form in advance, you will have to upload the form as part of your website – WebPlus gives you an option of doing this. However, it's more convenient for testing if you can preview your forms as you create them, and submit them for testing BEFORE uploading your site.

Adding a form in WebPlus

In the WebPlus interface select **insert/form** and choose either a **new blank form** or use the **form wizard** option



If you choose **new blank form** you will be returned to your WebPlus page where you will click to insert your form. When you see the blank form on your page, double click on the form to edit its **action** properties. If you choose the **form wizard option**, you will be taken through a few other steps to create your form. The final screen of the wizard will be the same **action** screen you will see if you double-click on a blank form.

Setting the action for your form

In the action dialog select the “**Remote Script**” option. Give a name to your form (*best practice is to keep the name in lower case letters with no spaces*) and then enter the URL of your remote script. (See illustration below)

Where should the form data be sent when the form is submitted?

No action An email address A script file from my hard drive **A remote script** Serif Web Resources

Form name: my_form

URL of remote script: http://yoursite.com/scripts/formmail.php

Method: ☒ POST ☐ GET

Encoding type: application/x-www-form-urlencoded

Target Window or Frame: Same Window _self

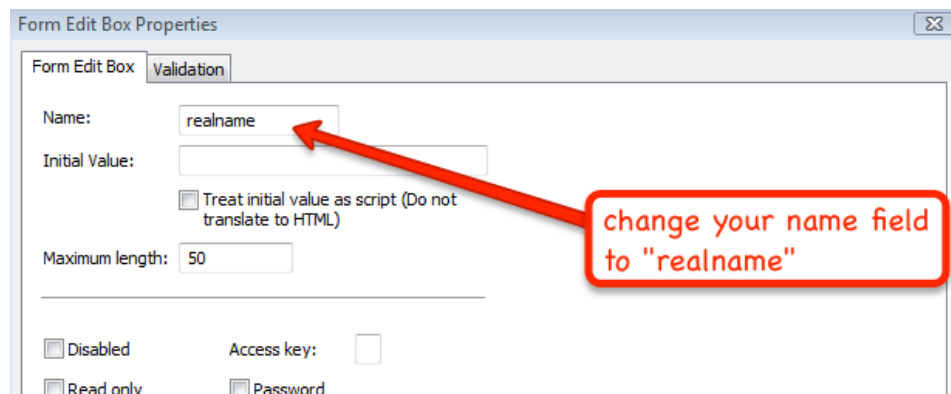
Char-set: UTF-8 (Default)

< Back Finish Cancel Help

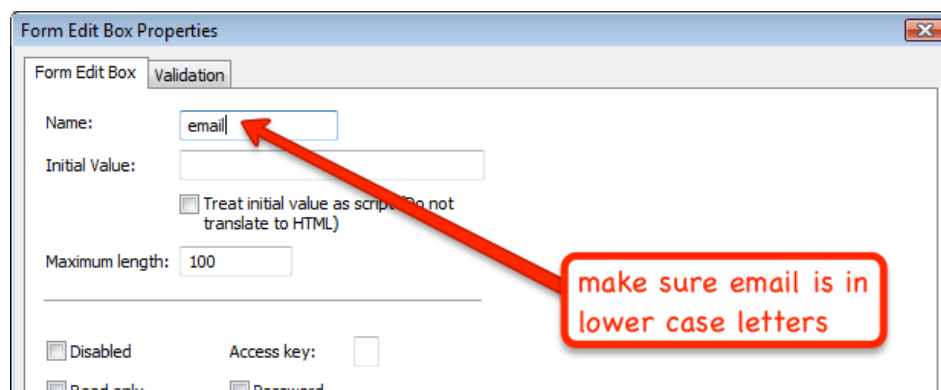
If you haven't yet uploaded your script, you can select the “**A script file from my hard drive option**” above. This will then give you the option of browsing for the script file on your computer. When you upload your site, the script file will be included as part of the upload.

The fields on your form

In order for your script to function perfectly with your forms, you **MUST** include two essential fields. One should be called “**realname**” and the other should be called “**email**”. On some of the WebPlus ready-made forms, you may have to rename some of the existing fields to reflect these requirements. You do this by double-clicking on an existing field name and editing its properties (*See illustration below*)



In the above example, we double-clicked on an existing field which WebPlus called **NAME** – we’ve changed it to read **realname** (*all lowercase letters*). Likewise, in the example below we’ve renamed the WebPlus supplied **EMAIL** name to **email** (*all lowercase*).



If you are creating a form from scratch, just be sure to create two fields with these names on your forms. Remember, the label of the form field can say something completely different e.g. Last Name or First Name can be the label for the **realname** field.

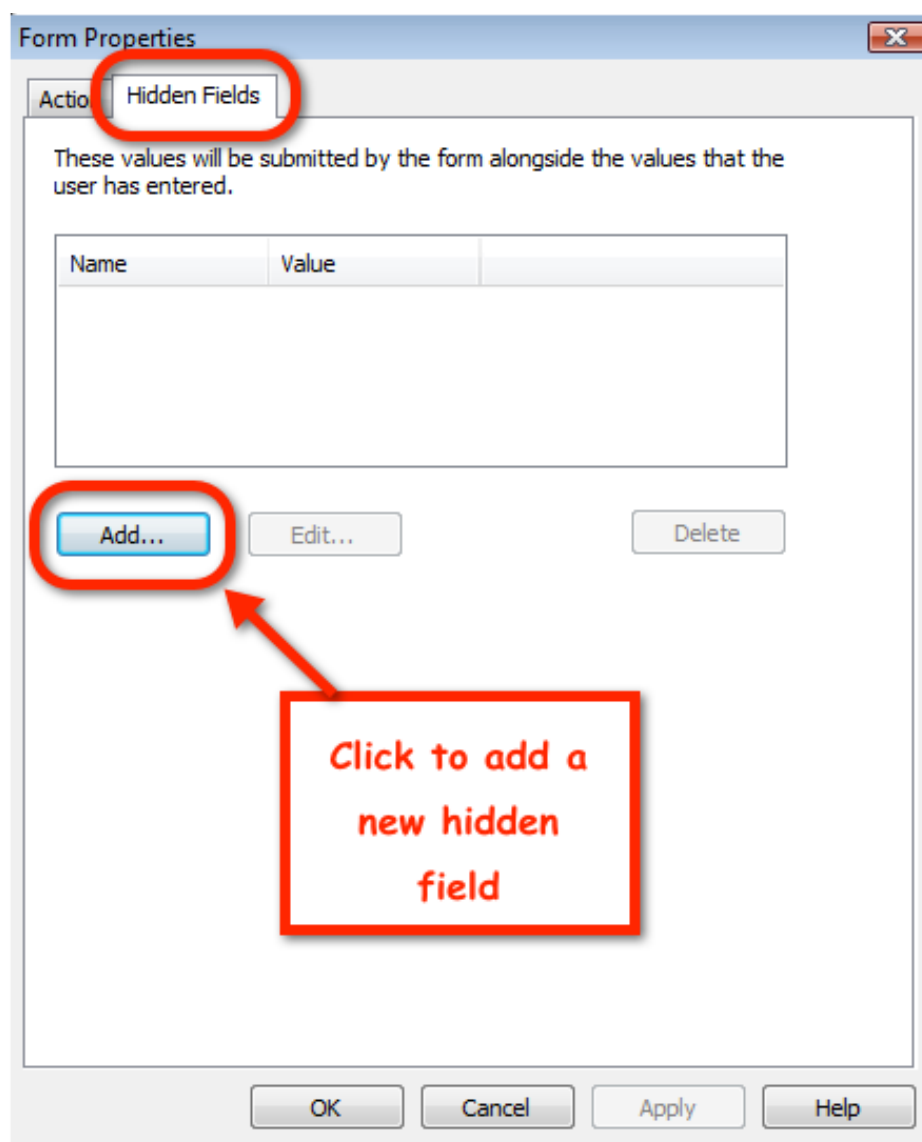
Once you’ve created your form, you’re ready to add some hidden fields to your forms to let Formmail know how to handle your form data. (*See next page*)

The Hidden Fields on your form

There are three basic things that your form needs to do when it's submitted to the script.

1. It needs to know where to send the form data.
2. It needs to know where to send a visitor upon SUCCESSFUL form submission
3. It needs to know where to send a visitor upon UNSUCCESSFUL form submission.

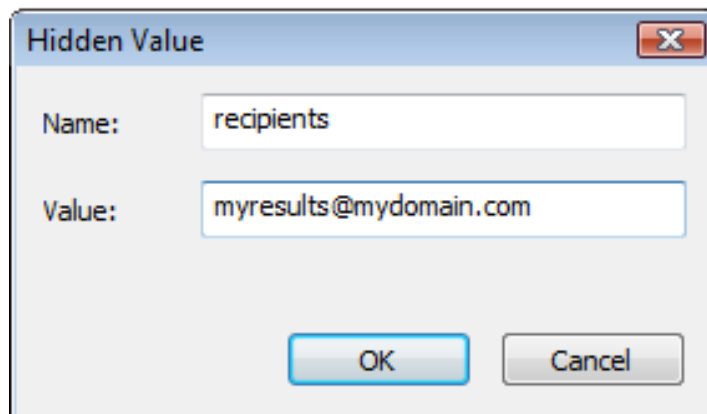
This information is fed to the script through invisible fields in your form. Start by double-clicking in a blank area of your form. This will bring up the Form Properties dialog again. Click on the Hidden Fields tab at the top of the dialog.



To add a new hidden field, click the **Add** button as illustrated above. A small pop-up window will appear where you can set up a hidden field name and a value. The illustration on the following page shows what field you need to create and the values you should give each field.

Recipients

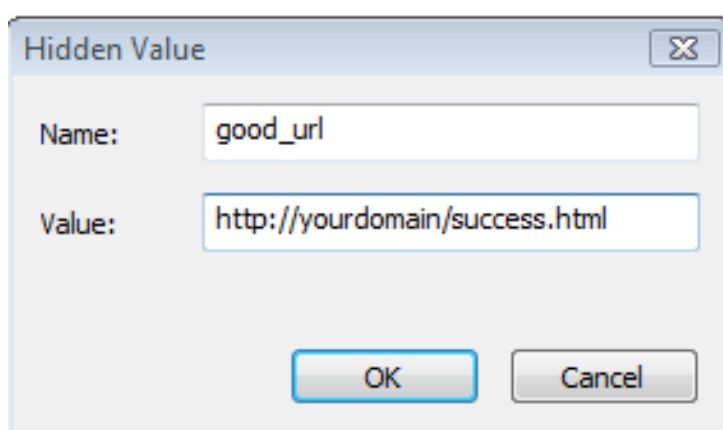
The first hidden field you need to create is the “**recipients**” field. This is where you enter the email address, or addresses, to where you want your form data sent.



A screenshot of a 'Hidden Value' dialog box. The title bar says 'Hidden Value' with a close button (X). Inside, there are two text input fields. The first is labeled 'Name:' and contains the text 'recipients'. The second is labeled 'Value:' and contains the text 'myresults@mydomain.com'. At the bottom of the dialog, there are two buttons: 'OK' and 'Cancel'.

Type the name **recipients** into the **name** field and the email address in the **value** field. Remember, the email address you enter here **MUST** match the email address (*or the domain part of the address*) that you set up in the **TARGET_EMAIL** parameter of your script. If you’ve set up the **AT_MANGLE** parameter in your script, replace the “@” symbol with the character string you defined in the script. You can add multiple email addresses using these same rules. Just be sure to separate each email addresses with a comma. Click **OK** to return to the form properties dialog and click **Add** to create another hidden field.

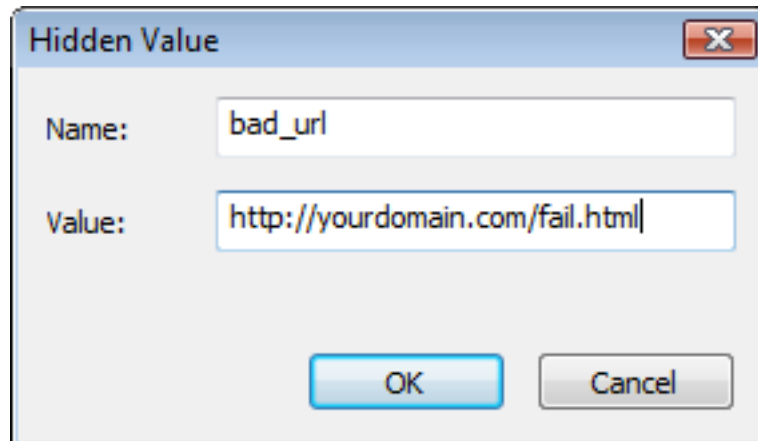
This time we need a hidden field to tell our script where to send visitors after a successful form submission. This would normally be a thank you page included in your website. Just enter **good_url** in the **name** box and the FULL URL of your success or thank-you page in the **value** box.



A screenshot of a 'Hidden Value' dialog box. The title bar says 'Hidden Value' with a close button (X). Inside, there are two text input fields. The first is labeled 'Name:' and contains the text 'good_url'. The second is labeled 'Value:' and contains the text 'http://yourdomain/success.html'. At the bottom of the dialog, there are two buttons: 'OK' and 'Cancel'.

Once the script has successfully processed a form, your visitor will be taken to your success page. You can create a number of different success pages for each form on your website. This allows you to “tailor” the success message to suit each form. Just enter the appropriate success page URL into the value box.

The last hidden field we need is one that tells the script where to send the visitor if there is a problem with processing the form data. This would typically be an HTML page in your website that advises visitors that there was a problem processing the form submission. It should apologise for the inconvenience and advise the visitor that the site administrator has been informed of the problem.



A screenshot of a 'Hidden Value' dialog box. The dialog has a title bar with the text 'Hidden Value' and a close button (X). Inside the dialog, there are two text input fields. The first field is labeled 'Name:' and contains the text 'bad_url'. The second field is labeled 'Value:' and contains the text 'http://yourdomain.com/fail.html'. At the bottom of the dialog, there are two buttons: 'OK' and 'Cancel'.

Enter **bad_url** in the **name** box and the FULL URL of the failure page in the **value** box. If a problem is encountered in processing the form, your script will automatically send you an alert email advising you of the problem, and divert your visitor to the failure page in your website. Just like with the **good_url** hidden field, you can have different failure pages for each of your forms, or you can use the same one for all forms.

The above are the bare minimum hidden fields you should have. The only one that is absolutely essential is the **recipients** field. Without this your form script won't be able to deliver the form results. The **good_url** and **bad_url** are optional, but should be included for good "netiquette". You could, if required, add another hidden field named **subject**. The value of this field would be set to a message you would want to see in the subject line of emails sent to your recipients e.g. **Contact Form Submission**.

Test your form

You can now preview your site in a browser by going to **file/preview site/preview site in xxx** (*xxx will be your browser of choice as listed in the preview site menu*). If your script and your success page have been uploaded to your server, you can test your form without uploading the whole site.

If there is a problem with your form submission, you will receive an alert email that gives you an indication of the error that has occurred. If your form was submitted and processed correctly, you will receive an email containing your form data.

If you get an error message, either on screen or via an alert message, you should go to the [Tectite Forum](#) and use the search box to find the error message you've received. You will then see how others have resolved the problem, or get a clearer indication of what may be wrong with your form or script.

Additional Information

This information is designed to help you enter your TARGET_EMAIL parameter to cater for different needs or to use double extension email addresses, such as **.co.uk**.

If your email address contains a double extension you should enter it as follows:

```
$TARGET_EMAIL = array(EMAIL_NAME."@bloggsdomain\co\.uk$");
```

Notice that any full-stop (*period*) in an email address MUST be preceded with a back-slash. The same rule applies if you are using a full email address that contains a full-stop (*period*) in the name. For example, if your email address was **george.bell@hotmail.com** it would be entered in the TARGET_EMAIL parameter as:

```
$TARGET_EMAIL = array("^george\.bell@hotmail\.com$");
```

You can add multiple domains or email address into the TARGET_EMAIL parameter. This is useful if you think you will be creating forms that could go to any one of a number of different email addresses. The correct format for adding multiple domains is as follows:

```
$TARGET_EMAIL = array(EMAIL_NAME."@rootsoftware\.com$",  
    EMAIL_NAME."@rootsoftware\.com\.au$",  
    EMAIL_NAME."@ttmaker\.com$",  
    EMAIL_NAME."@timetabling\.org$",  
    EMAIL_NAME."@timetabling-scheduling\.com$",  
    EMAIL_NAME."@tectite\.com$"  
);
```

The above will allow forms to be sent to any email account on any of the domains listed. Note: this doesn't mean that emails WILL be sent to all domains – it simply means that you can define recipients in the **recipients hidden fields** on your forms that match with any, some or all of these domain names.

Likewise, you can add several SPECIFIC email addresses that you may want to use as recipients in your form's **recipients hidden field**. These should be entered in the TARGET_EMAIL parameter as follows:

```
$TARGET_EMAIL = array("^russell\.robinson@rootsoftware\.com$",  
    "^info@ttmaker\.com$",  
    "^sales@timetabling\.org$",  
    "^webmaster@timetabling-scheduling\.com$"  
);
```

Further Reading

This guide has explained how to set up the Formmail script to perform the basic task of receiving form results to one or more email addresses. The script is capable of much more, including generating HTML success pages containing the form data that visitors submit, storing information in a database or CSV file and renaming and storing files uploaded through your forms. You can also use Formmail to process multi-part forms, or send forms to recipients based upon selection boxes in your forms.

To learn more about these and many other features of this script, please visit the [Tectite Website](#) and check out the [tutorials](#) and the [Index of All Settings](#). If you get to grips with all the features this script offers, you'll never look at third-party hosted solutions again.