

# Form Mail Script Documentation

Thank you for your interest in our Form Mail Script. We recommend that you read this document carefully.

<b>1</b>	<b>Introduction .....</b>	<b>2</b>
<b>2</b>	<b>Installation of the script.....</b>	<b>2</b>
2.1	Step by Step .....	2
2.2	Include the Script .....	4
2.3	Upload the script to the server .....	5
2.4	File and Folder Permissions.....	5
<b>3</b>	<b>Set up and configuration .....</b>	<b>5</b>
3.1	Options in “index.php” .....	5
3.2	Attachments .....	8
3.3	Captcha.....	10
3.3.1	Send E-mails with SMTP .....	11
<b>4</b>	<b>HTML template customization - /templates/form.tpl.html .....</b>	<b>12</b>
4.1	Form Fields .....	12
4.1.1	Select Menus .....	13
4.1.2	Checkboxes .....	14
4.1.3	Radio Buttons .....	14
4.1.4	Required Form Fields .....	15
4.1.5	E-mail Syntax.....	15
4.1.6	Compare Two or More Fields.....	15
4.2	Advanced Error Messages.....	16
4.2.1	{required:fieldname='Error message':endrequired} .....	17
4.2.2	{syntax:fieldname='Error message':endsyntax} .....	17
4.2.3	{error:fieldname='Default text'  ='Error message':enderror}.....	18
4.3	Thank You Page .....	18
4.4	Include Files.....	18
4.5	Ad Link Format.....	19
4.6	Define HTML Template .....	19
4.7	Define Mail Template .....	20
<b>5</b>	<b>E-mail Template Customization - /templates/mail.tpl.txt .....</b>	<b>20</b>
5.1	Mail Header.....	20
5.2	Mail Body .....	21
5.3	Auto Responder – Confirmation e-mail .....	21
5.4	Display all form data with one placeholder.....	21
5.5	Environment variables and server date and time .....	22
<b>6</b>	<b>Language files .....</b>	<b>22</b>
<b>7</b>	<b>The Log File - log/logfile.txt.....</b>	<b>23</b>
<b>8</b>	<b>Support .....</b>	<b>23</b>
<b>9</b>	<b>License Information .....</b>	<b>23</b>

# 1 Introduction

Form Mail Script allows you to place many different kinds of forms on your website. Website visitors can fill in these forms and submit them. The script puts the data from the form in an e-mail (text and HTML) and sends them to a pre-defined e-mail address. The script also provides some security features to reduce the possibility of unauthorized use.

Once installed, you can access the script either directly by calling the URL to the script in your browser or by pointing to the script from an existing form on your website.

The script is template driven which means you can fully customize the layout of the form page and the e-mail. You can run an unlimited number of forms with one script, send attachments and confirmation e-mails.

List of features:

- HTML and plain text e-mails
- Unlimited forms
- E-mail harvesting protection
- Upload files send as attachments
- Field validation with error messages
- Confirmation e-mail (auto responder)
- Log file
- Captcha
- IP blocker
- Referrer check
- Redirect to a thanks page
- Redirect to an error page
- E-mail preview
- Can be included by other PHP files

The included default HTML template is written to the specifications of XHTML 1.0. Therefore, HTML tags like `<br>` or `<input>` contain an ending slash (`<br />` or `<input type="text" name="field" />`). If you are using an existing HTML form, you do not need to add the ending slash.

## 2 Installation of the script

The script is almost ready to use.

### 2.1 Step by Step

Assuming that you already have unpacked the script files these step-by-step instructions

## Step 1

Open the file `/templates/mail.tpl.txt` in a text editor. Enter your e-mail address in the head of the file where it says `To`:

```
To: your-e-mail@example.com
```

Replace `your-e-mail@example.com` with your own e-mail address.

You can also add your name, but it is important that you enclose the name by double quotes (`"`) and the e-mail by angle quotes (`<` and `>`):

```
To: "First name Last name" <your-e-mail@example.com>
```

You can now save the file and close it.

## Step 2

Create a subfolder on your server (i.e. `/mail/` or `/contact/`). Copy all script files into the new folder on your server. It is important that you upload all files, folders and sub-folders to your server. It is also important that the original file and folder structure remains intact. See also the chapter [Upload the script to the server \(2.3\)](#).

## Step 3

After you have uploaded the files you can call the URL of the script in your browser, i.e.:

```
http://www.example.com/contact/index.php
```

The script is now ready for you to test. Fill in the fields and send the form.

## Step 4

Check your e-mail account for new mails. You should be a bit patient. It may take a couple minutes until the e-mail arrives into your e-mail account.

Should you not receive any e-mails even after quite some time, start again with step 1 and try several different e-mail addresses.

In case that is not successful either, you should test if your server is capable of sending e-mails at all. Create a PHP file `mailtest.php` with following content:

```
<?php
mail('your-e-mail@example.com', 'Test Subject', 'Test Message');
?>
```

Upload that file to your server and call the URL of it in your browser. In the rare event that this does not work, you should contact your web host.

## 2.2 Include the Script

You can run the script as stand-alone and customize the layout using the HTML template. If that is the case, you can skip this chapter.

You can include the script in other PHP files so that it fits into an existing website and matches the layout. Here is what you do:

### Step 1

You can use the PHP statement `include` to include the script at the very top (the first line of the file, above every other character, with no character in front of it) of the file:

```
<?php include 'index.php'; ?>
```

### Step 2

Go to the place in the file where you want the form to appear. Add following line of code:

```
<?php echo $f6l_output; ?>
```

### Step 3

It is **important** that you delete this line of code from the file `index.php`. Otherwise the form will appear on top of your page. Look for this line at the bottom of the file `index.php`:

```
echo $f6l_output;
```

You can also change it into a comment by adding two slashes in front of the line, like so:

```
//echo $f6l_output;
```

### Step 3

Since you are including the script, you probably won't need the HTML head structure. Just strip it from the HTML template `form.tpl.html` and leave everything between `<body>` and `</body>` intact.

**Done.**

## 2.3 Upload the script to the server

All files have to be uploaded in text mode (ASCII mode) except for the font file(s) in the folder `/inc/font/`. The font file `daft.ttf` has to be transferred in binary mode.

Most FTP programs have the option to set the mode to Automatic. With the mode set to Automatic you won't have to worry about it.

## 2.4 File and Folder Permissions

Most, if not all, FTP programs allow you to change the file permissions. That can either be done with a command line and the command `chmod` or with some clicks on menus and checkboxes. We have published a guide with examples of common FTP programs on our website.

<http://www.stadtaus.com/en/tutorials/chmod-ftp-file-permissions.php>

# 3 Set up and configuration

In the script root folder (the folder that contains all the other folders like `/docu/`, `/inc/`, `/languages/`, `/templates/` etc.) you can find the file `index.php`. That file is basically the configuration file. You can open the `index.php` file in a text editor.

## 3.1 Options in “index.php”

```
$referring_server = '';
```

This variable contains a comma separated list of domains that are allowed to link to the script.

For example:

```
$referring_server = 'www.example.com, example.com, 200.77.100.166';
```

Please note: Enter only domain names or IP addresses. Do not enter whole URLs or paths. There can't be any `http://` or slashes in there.

If you forget to add your own domain or someone else is pointing to your form from another website, following error message will be displayed:

```
Wrong or empty referrer (referring site)!
```

Visitors who have a firewall software installed on their computer may also see get this error message because some firewalls block the referrer. You can work around this by leaving the variable `$referring_server` empty. This will disable the referrer check.

If you do so, make sure you place your e-mail address directly in the e-mail template. This makes the script useless to spammers, because e-mail can only be sent to the address in the template.

`$language`

This variable lets you choose the language the script will use for error messages. Take a look into the folder `/languages/` in order to find out which languages are available.

`$ip_banlist`

This variable can contain a comma separated list of IP addresses that want to ban from accessing the script. Be careful, many different users may use a single IP address. If you do not want to use this feature, just leave the variable empty.

`$ip_address_count`

This variable lets you set the maximum number of e-mails that a user can send during one visit to your web site. Once this number is reached, the user will not be able to send another e-mail from your site until the next time he dial in to the Internet. The number 0 (null, zero) turns this function off.

When you use this feature, the logging function is also turned on. This is necessary so that the script will be able to keep track of the IP addresses of the visitors that are sending mail from your web site. In order for this to work, the directory `/logfile/`, and the file `/logfile/logfile.txt` both must have their writable permissions set to `chmod 777`. For more information about file and folder permissions see chapter File and Folder Permissions (2.4).

`$ip_address_duration`

This variable sets the number of hours, from the time someone firsts sends an e-mail from your site, before they can send more e-mails. After the number of hours you set pass, the user can send e-mails from your site again.

`$show_limit_errors`

This variable lets you decide if you want the script to show an error message, once the user has reached the maximum number of e-mails you have set.

`$log_messages`

This variable lets you turn the logging function on or off. In order for this feature to work, the directory `/logfile/`, and the file `/logfile/logfile.txt` both must have their writable permissions set to: `chmod 777`. For more information about file and folder permissions see chapter File and Folder Permissions (2.4).

`$text_wrap`

This variable sets the maximum number of character you want on a line in your e-mail. After this number is reached, a line break will automatically be added. If no value is defined, no line break will be added.

`$show_error_messages`

This variable is initially set to `yes`. This will let the script display you error messages during the configuration process. This will help you to install the script. After you have finished configuring the script, you should set this variable to `no`. Do not leave it set to `yes`. It is not wise to give visitors to your site a view into the workings of the script.

`$attachment`

Please find detailed instructions about file uploads and attachments in chapter [Attachments \(3.2\)](#).

`$captcha`

Please find detailed instructions about file uploads and attachments in chapter [Captcha \(3.3\)](#).

`$path['logfile']`

This variable contains the path to the log file.

`$path['templates']`

This variable contains the path to the HTML and text mail template files. The path can be **relative** (`./templates/` or `templates/`) or **absolute** (`/usr/local/etc/httpd/.../form_mail/templates/`). Please be sure to include the ending slash.

`$file['default_html']`

If you do not specify an HTML template in the form, this variable contains the file name of the HTML template that the script uses by default. The script will automatically use this template.

`$file['default_mail']`

If you do not specify a mail template in the form, this variable contains the file name of the mail template that the script uses by default. The script will automatically use this template.

```
$add_text
```

You can add further values, text, variables or other content to this variable. This content can then be displayed in both the HTML and mail template by using markers/placeholders within curly brackets { and }.

If you want to add your own values to the variable `$add_text` please follow the pattern from the example entries.

```
$add_text = array(
    'txt_additional' => 'Additional',
    'txt_more'       => 'More'
);
```

The first part of each entry is the name of the placeholder. To use the value of `txt_addition` in your template, place the name of the entry inside curly brackets.

Example: `{txt_addition}`.

The name of an entry must not contain white spaces or special characters. Only the underscore (`_`) is allowed.

## 3.2 Attachments

The script allows you to send an arbitrary number of attachments. In order to be attached to an e-mail, a file would have to be uploaded by the form user beforehand. If you want to use this feature, please follow these steps:

### Step 1

Copy the file `form.tpl.html` from the folder `/templates/example_upload_attachment/` to the folder `/templates/`. That will replace the existing `form.tpl.html`.

The new `form.tpl.html` contains following additional form field:

```
<input type="file" name="my_attachment" />
```

You are free to change the name of the field or to add more fields of the type `file`. If you have more than one upload field, please make sure each field has a different name (`name=""`).

Furthermore, it is important that the HTML tag `<form>` contains following attribute:

```
enctype="multipart/form-data"
```

It would look like this:

```
<form action="{script_self}" method="POST"  
enctype="multipart/form-data">
```

Otherwise, the script won't send the attachments.

The mail template does not require any changes.

## Step 2

Please make sure that the folder `/temp/` is writable (`chmod 777`). For more information about file and folder permissions see chapter File and Folder Permissions (2.4).

## Step 3

Activate the attachment feature by setting the variable `$attachment = 'yes'` in `index.php`.

## Done.

Following additional options can be set:

```
$attachment_files
```

This variable contains a comma separated list of file types that are allowed to be sent as attachment.

Please note: You should be very careful about what file types you allow for upload. Please bear in mind that in the end you will receive those files in your inbox.

```
$attachment_size
```

This variable contains the maximum size of an attachment in bytes. The maximum attachment size is applied for each uploaded file. For example: You have set a maximum attachment size of `1024 Kbytes` and have three upload fields. Each of the three uploaded files can be `1024 Kbytes` which results in `3072 Kbytes` for all three files together.

Please note: Upload fields cannot be set as required fields.

Earlier versions of the script did not allow the combination of attachments and HTML formatted e-mails. As of version 3.2 this is no longer the case.

### 3.3 Captcha

The Captcha feature displays an image with a string of characters. In order to prove that the visitor is not an automatic spam program, he has to enter the characters shown in the image into a form field. In case both strings match the comment will be accepted.

Note: The Captcha feature requires GD2 library and Freetype to be installed on the server which is the case on most servers. Otherwise, an error message will be displayed or the image does not contain any text.

In order to use this feature, follow these steps:

#### Step 1

Copy the file `form.tpl.html` from the folder `/templates/example_captcha/` to the folder `/templates/`. That will replace the existing `form.tpl.html`.

The new `form.tpl.html` contains following additional lines:

```
{txt_captcha_note}  
<br /><br />  
{captcha_content}
```

The placeholder `{txt_captcha_note}` contains the instructions for the user. The placeholder `{captcha_content}` will be replaced by the HTML code of the Captcha image and the form field.

#### Step 2

Please make sure that the folder `/temp/` is writable (`chmod 777`). For more information about file and folder permissions see chapter File and Folder Permissions (2.4).

#### Step 3

Activate the Captcha image by setting the variable `$captcha = 'yes'` in `index.php`.

**Done.**

Please note: Even though the Captcha feature prevents the form from being automatically submitted, there are enough spammers who work the form manually. Therefore, it is possible that you still receive spam messages send by the form.

Though the Captcha feature is fully functioning, it has some limitations. Currently, there is no way to control color or size of the box or of the letters.

### 3.3.1 Send E-mails with SMTP

If you don't know whether you should use SMTP, skip this chapter. Using SMTP is only necessary, if the internal PHP function mail() cannot be used (i.e. on a Windows server).

In order to send e-mails using SMTP you need an e-mail account (Pop3).

#### Step 1

Open the file `/inc/formmail.inc.php` in a text editor and look for this part:

```
$configuration['mail_type']      = 'mail'; // (mail, smtp)
$configuration['mail_from']      = 'postmaster@' .
$_SERVER['SERVER_NAME'];

$configuration['smtp']['host']   = 'example.com';
$configuration['smtp']['port']   = 25;
$configuration['smtp']['helo']   = $_SERVER['SERVER_NAME'];
$configuration['smtp']['auth']   = true;
$configuration['smtp']['user']   = '';
$configuration['smtp']['pass']   = '';
```

#### Step 2

Replace

```
$configuration['mail_type']      = 'mail';
```

with

```
$configuration['mail_type']      = 'smtp';
```

#### Step 3

Replace the host name `example.com` with the host name of your e-mail account.

```
$configuration['smtp']['host']   = 'example.com';
```

## 4. Schritt

Enter here the user name and password of your e-mail account.

```
$configuration['smtp']['user'] = '';  
$configuration['smtp']['pass'] = '';
```

Done

.

## 4 HTML template customization - /templates/form.tpl.html

The folder `/templates/` contains the files that determine the layout of your page. You can use HTML and CSS to customize the layout as you wish.

If you are using a WYSIWYG editor like Dreamweaver, Frontpage or NetObjects, please make sure that the software does not make any changes by itself.

The subfolder `/examples/` contains additional HTML forms. You can use them as examples for creating your own forms.

The folder `/templates/` contains an `.htaccess` file to protect your template files from being accessed via browser. Only the script and you (via FTP) have access to the template files.

Once you have finished working on the HTML templates it is advised to run the script (not the HTML template – the whole script) through the HTML validator of the W3 consortium. The validator helps you find improper coded HTML that may prevent the script from working.

<http://validator.w3.org/>

Please note: Changing the extension of the template file from `.html` to `.php` has no effect. The file will not be recognized as a PHP file and therefore no PHP code will be parsed. Actually, the extension doesn't matter. It could be `.html`, `.txt`, `.tpl`, `php` or no extension at all. If you want to include PHP code in the template, please use the include feature (chapter 4.4).

### 4.1 Form Fields

For each form field the script will automatically create a placeholder with the same name as the form field. That comes in handy if an error occurs (i.e. the user forgot to fill in a required field) and the form appears again, the script can pre-populate the form fields so the user doesn't have to fill in all fields again. For example:

```
<input type="text" name="lastname" value="{lastname}">
```

The script will replace `{lastname}` with the value that the user entered before he submitted the form.

You can add as many fields to a form or change the fields of the example forms that come with the script. Please make sure that the field names and the placeholder names do not contain white spaces, or special characters. They may have underscores (`first_name`), though.

### 4.1.1 Select Menus

The script can process select menus like following:

```
<select name="" size="">
  <option value=""></option>
  <option value=""></option>
</select>
```

The script can also process multiple select fields. You can find an example in the HTML template `/templates/example_multiples/form.tpl.html`.

In order to make sure that the user's selection appears after submitting the form for preview, you can use a special placeholder.

```
<select name="Salutation" size="2">
  <option value="Mr" {select:Salutation=Mr}>Mr</option>
  <option value="Ms" {select:Salutation=Ms}>Ms</option>
</select>
```

The placeholder `{select:Salutation=Ms}` consists of the following elements. In the first position is the type of form field you are using. In this case it is a select menu (= `select`). After that follows a colon (:). In the second position is the name of the form field. In the example above that is `Salutation`. After that an equal sign (=) follows. At the last position comes the value of the option tag (`value=""`).

To make the selected value appear within the mail template, you need to enter the field name surrounded by curly brackets { } in the mail template, i.e. `{Salutation}`

In case a user did not select one of the options the script removes the placeholder from the mail template.

If you want a select field to be a required field and you have listed it in the `required_fields` form field, the first option of the list has to have an empty value. Otherwise the script will not show an error.

```
<select name="Salutation" size="2">
  <option value="">Please select</option>
  <option value="Mr" {select:Salutation=Mr}>Mr</option>
  <option value="Ms" {select:Salutation=Ms}>Ms</option>
</select>
```

You can find more examples in the template  
`/examples/advanced_form.tpl.html`.

### 4.1.2 Checkboxes

The script can process checkbox fields like following:

```
<input type="checkbox" name="" value="" />
```

The script can also process multiple checkbox fields. You can find an example in the HTML template `/templates/example_multiples/form.tpl.html`.

In order to make sure that the user's selection appears after submitting the form for preview, you can use a special placeholder.

```
<input type="checkbox" name="Newsletter" value="Yes"
{checkbox:Newsletter=Yes} />
```

The placeholder `{checkbox:Newsletter=Yes}` consists of following elements. In the first position is the type of form field you are using. In this case it is a checkbox field (= `checkbox`). After that a colon (`:`) follows. In the second position is the name of the form field. In the example above that is `Newsletter`. After that follows an equal sign (=). In the last position is the value of the option tag (`value=""`).

In order for the selected value to appear in the mail template, you need to enter the name surrounded by curly brackets `{ }` in the mail template, i.e.

```
{checkbox:Newsletter}.
```

You can find more examples in the template  
`/examples/advanced_form.tpl.html`.

### 4.1.3 Radio Buttons

The script can process radio button fields like following:

```
<input type="radio" name=" " value="" />
```

In order to make sure that the user's selection appears after submitting the form for preview, you can use a special placeholder.

```
<input type="radio" name="Salutation" value="Mr"
{radiobutton:Salutation=Ms} />
```

The placeholder `{radiobutton:Salutation=Ms}` consists of the following elements. In the first position is the type of form field you are using. In this case it is a

radio button field (= `radiobutton`). After that follows a colon (:). In the second position is the name of the form field. In the example above that is `Salutation`. After that follows an equal sign (=). In the last position is the value of the option tag (`value=""`).

In order to make the selected value appear in the mail template, you need to enter the name surrounded by curly brackets - { } - in the mail template, i.e. {`Salutation`}.

You can find more examples in the template  
`/examples/advanced_form.tpl.html`.

#### 4.1.4 Required Form Fields

Every form field can be marked as a required field. This means that the e-mail will not be sent until these fields are filled in. Enter a comma separated list of field names that you want to be required into a hidden form field with the name `required_fields`. For example:

```
<input type="hidden" name="required_fields" value="lastname,  
email, subject" />
```

This will make the form fields `lastname`, `email` and `subject` required fields.

#### 4.1.5 E-mail Syntax

Like the required fields you can define fields that you want to be checked for correct e-mail syntax. Enter a comma separated list of field names that you want to be checked in a hidden form with the name `email_fields`. For example:

```
<input type="hidden" name="email_fields" value="email_1,  
email_2" />
```

#### 4.1.6 Compare Two or More Fields

The script allows you to compare the content of two or more fields. An error message will be displayed if the content is not identical. Possible use would be for confirming an e-mail address or password. Put following hidden field to your form:

```
<input type="hidden" name="compare_fields[]" value="email,  
confirm_email" />
```

The line above would tell the script to test whether the fields `email` and `confirm_email` contain the same value.

The number of different sets of fields the script can check is not limited.

```
<input type="hidden" name="compare_fields[]" value="email,
confirm_email" />
<input type="hidden" name="compare_fields[]" value="password,
confirm_password" />
```

## 4.2 Advanced Error Messages

If a required field has not been filled in by the user, an error message will be displayed above the form fields. This is the default setting.

The script allows you to display error messages right beside, above or below a form field. Moreover, the script allows you to alter the appearance of the form field label.

This screenshot shows how that could look like:

**Form Mail Script**

Please fill in the following form fields:

Last name  
E-mail  
Subject

First name

Last name \*   
Please enter your last name.

E-mail \*   
Please enter your e-mail address.

Subject \*   
Please enter a subject.

Comments

\* = Required fields

You can see both the default message above the form and the advanced error messages that show up above each form field.

In order to implement advanced error messages, following three placeholders are available:

```
{required:fieldname='Error message':endrequired}

{syntax:fieldname='Error message':endsyntax}

{error:fieldname='Default text' ||= 'Error message':enderror}
```

#### 4.2.1 {required:fieldname='Error message':endrequired}

Let us suppose you have a field `last_name` and you define it as required field. You can use following variable for the error message:

```
{required:last_name='Please enter your last name.':endrequired}
```

In case the visitor has not filled in the field, the variable will be replaced with the error message:

Please enter your last name.

You could format the error text using CSS:

```
{required:last_name='<span style="font-weight:bold;color:#FF0000;">Please enter your last name.</span>':endrequired}
```

The error text now appears in bold font and in red color:

#### 4.2.2 {syntax:fieldname='Error message':endsyntax}

Let us suppose you have a field `email` and you define it as e-mail syntax field. You can use following variable for the error message:

```
{syntax:email='Please enter a valid e-mail address.':endsyntax}
```

In case the visitor has not filled in the field correctly, the variable will be replaced with the error message:

Please enter a valid e-mail address.

You could format the error text using CSS:

```
{syntax:email='<span style="font-weight:bold;color:#FF0000;">Please enter a valid e-mail address.</span>':endsyntax}
```

The error text now appears in bold font and in red color.

### 4.2.3 {error:fieldname='Default text' ||= 'Error message':enderror}

You can use following variable if you want to change the appearance of the form field label. Let us suppose you have a field `subject` and you define it as required field. You can use following variable in order to change the appearance of the label:

```
{error:subject='Subject' ||= '<span style="color:#FF0000;">Subject</span>.' :enderror}
```

Without an error the first part (everything before the two vertical lines - pipes) will be displayed. In case the visitor has not filled in the field, the variable will be replaced with the second part:

For a better understanding you could take a look at the example in the folder `/templates/examples_error_messages/`.

## 4.3 Thank You Page

After the form has been submitted the user will be presented with a confirmation message and the data he just entered in the form. You can change this and let the user be redirected to a special thanks page. You can define the URL of that page in a hidden form field with the name `thanks`. For example:

```
<input type="hidden" name="thanks" value="http://www.example.com/thanks.html" />
```

Please enter the whole URL including `http://` and the domain name (i.e.: `http://www.example.com/`).

If you don't define this field or leave it empty the script displays the data entered by the user.

## 4.4 Include Files

The script allows you to include files directly into the HTML template files. It makes no difference whether you want to include PHP, HTML or other text files. For example:

```
<INCLUDE FILENAME="header.php">
```

The script automatically replaces this line with the content of the file `header.php`. Please pay attention that relative paths (`../` etc.) start at the main directory of the script (same level as the `index.php` file).

If you use absolute paths (path starts with slash), the path starts at the `www` root folder of your web account by default. For example:

```
/logs/  
/images/
```

```
/layout/  
/layout/header.php  
/form_mail/  
/form_mail/templates/  
/form_mail/templates/form.tpl.html  
/form_mail/index.php
```

Let us assume that you want to include the file `header.php` into the file `form.tpl.html`. If you use a relative path, you need to start at the same level as the file `index.php` is:

```
<INCLUDE FILENAME="../../layout/header.php">
```

If you want to use an absolute path, you need to write your path starting at the `www` root folder:

```
<INCLUDE FILENAME="/layout/header.php">
```

You can find examples about including in the folder `/templates/examples_includes/`.

## 4.5 Ad Link Format

You can use CSS to format and position the link. You can address the link using the CSS id `poweredby`. The HTML template `form.tpl.html` contains following section in the head part:

```
<style type="text/css">  
<!--  
  
    #poweredby {  
        text-align:center;  
        margin-top:30px;  
    }  
  
-->  
</style>
```

This CSS code centers the link and puts a margin of 30 pixels above it.

## 4.6 Define HTML Template

If you are calling the script from an existing form in a HTML page and you do not want to use the default HTML file, you can define a different HTML file in a hidden form field with the name `html_template`. For example:

```
<input type="hidden" name="html_template" value="form.tpl.html"  
>
```

## 4.7 Define Mail Template

The mail template can be defined the same way. You have the option to define a single mail template. For example:

```
<input type="hidden" name="mail_template" value="mail.tpl.txt" />
```

You can also define more than one mail template. Each of these templates can contain different recipients and content. For example:

```
<input type="hidden" name="mail_template" value="mail.tpl.txt, mail2.tpl.txt" />
```

Please note: This is the way to set up a confirmation e-mail (auto responder). Read more about that in the chapter Auto Responder – Confirmation e-mail (5.3).

## 5 E-mail Template Customization - /templates/mail.tpl.txt

### 5.1 Mail Header

At the top of the mail template you can find the lines of the mail header. For example:

```
Recipient      To:
Sender         From:
Subject       Subject:
Carbon Copy    CC:
Blind Carbon Copy BCC:
```

You can define the recipient of an e-mail like this:

```
To: name@example.com
```

In addition, you can add several recipients with carbon copy and blind carbon copy:

```
Cc: mail@example.com, chef@example.com
Bcc: secret@example.com
```

Please note: If you have set your e-mail account as “Catch all”, you can not send more than one e-mail to one domain. If more than one e-mail is sent to a domain, only one will be received. The following example will not work:

```
To: name@example.com
```

Cc: chef@company.com

You can define the sender within the mail template. For example:

```
From: chef@example.com
```

You can also use information from the form. For example:

```
From: {email}
```

You also can add the name of the sender. In this case the name is followed by the e-mail address which stands between angle brackets.

```
From: "{firstname} {lastname}" <{email}>
```

## 5.2 Mail Body

After the e-mail head area (header) an empty line (two line breaks) follows. Below that line follows the mail content. You can customize the mail content as you like. If you add fields to your form, you will also need to place the name of those fields as placeholders in the mail template.

Make sure that the names of the fields in your form are identical to the placeholders in your mail template.

## 5.3 Auto Responder – Confirmation e-mail

You can create an auto responder message to the person who submitted the form by creating another mail template. The To: line of the new template contains – instead of your own e-mail address – the e-mail address that the user supplied. You can find an example in the `/templates/examples/mail_auto_responder.tpl.txt`.

The definition of the mail template looks like this:

```
<input type="hidden" name="mail_template" value="mail.tpl.txt,  
mail_auto_responder.tpl.txt">
```

The definition in `index.php`:

```
$file['default_mail'] = 'mail.tpl.txt,  
mail_auto_responder.tpl.txt ';
```

## 5.4 Display all form data with one placeholder

You can display all form data with just one placeholder. Add following placeholder to the e-mail template:

```
{all_content}
```

If you have your mail template formatted as HTML e-mail, you can use following placeholder.

```
{all_content_table}
```

This placeholder shows all form field values structured in an HTML table.

## 5.5 Environment variables and server date and time

The following environment variables are provided by Form Mail Script. In order to make these variables appear in an e-mail you need to put following placeholders in your template:

<code>{env_user_agent}</code>	Browser software of the user (Mozilla, Netscape, Opera, Internet Explorer)
<code>{env_remote_address}</code>	IP-Address of the user
<code>{env_remote_host}</code>	Host name of the user
<code>{env_year}</code>	Year
<code>{env_month}</code>	Month
<code>{env_day}</code>	Day
<code>{env_hour}</code>	Hour
<code>{env_minute}</code>	Minutes
<code>{env_second}</code>	Seconds

## 6 Language files

The template files contain markers with { and } like: `{txt_preview}`. These markers will be replaced by its counterparts from the language file `/languages/language.en.inc.php`.

You are free to edit that file and change the words and phrases. You can even add HTML code in order to format the text. You are also free to replace the markers in the template files with real words, or to put those markers in other positions within the template file. You can also enhance the language file using the given pattern.

The following placeholders are not included in the language file and should not be changed.

```
{message}  
{fields}
```

On a side note: If you have translated a file into your language, it would be awful nice if you could send us the translated file. Thanks in advance :-)

## 7 The Log File - log/logfile.txt

The folder `/logfile/` and the file `logfile.txt` must be writeable (`chmod 777`). For more information about file and folder permissions see chapter File and Folder Permissions (2.4).

The folder `/logfile/` contains an `.htaccess` file to protect your log file from being accessed via browser. Only the script and you (via FTP) have access to the log file.

## 8 Support

For the answers to your questions about Form Mail Script, and for the solutions to your problems with Form Mail Script, visit our support forum.

<http://www.stadtaus.com/forum/>

## 9 License Information

The script is free for private usage and non-profit organizations. The free version requires the ad link. If you want to use the script without the ad link, you need to purchase a license.

Purchasing a license is also required for:

- Companies
- Websites with ads or any other revenue stream
- Adult websites

Once you have purchased a license, we will send you a new license file which will replace the existing `config.dat.php` file in the folder `/inc/`. The new file removes the ad link, but does not interfere with the changes you have made to the script. You don't have to start over.