

Formmail

User Guide



© April 6, 2005 Linux Web Host. All rights reserved. The content of this manual is furnished under license and may be used or copied only in accordance with this license. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, recording, or otherwise, without prior written permission of Linux Web Host.

Doc ID: FORMAIL-002.4 (4/6/05)

Contents

Getting Started	Chapter 1	
	Overview	1
	Required files	1
Configuration	Chapter 2	
	Overview	2
	Default configuration file	2
	Using File Manager to edit the configuration file	4
	Using SSH to edit the configuration file	5
	Uploading the configuration file via FTP	6
Form Structure	Chapter 3	
	Overview	7
	The form itself	7
	Getting information from users	7
	Submitting the form	8
	Sample Form	8
Input Fields	Chapter 4	
	Overview	10
	recipient	10
	subject	10
	email	10
	realname	10
	sort	11
	redirect	11
	required	11
	env_report	11
	title	12
	return_link_url	12
	return_link_title	12
	background	13
	bgcolor	13
	text_color	13
	link_color	13
	vlink_color	13
	alink_color	14

Chapter 1: Getting Started

Overview

Formmail is an application designed to allow your users to email you from your website, without having to open a separate email client, such as Outlook, Thunderbird, or Eudora. All your users will have to do is fill out the form you provide and click a button.

This guide provides complete instructions regarding the HTML required to actually create a form, but instructions regarding the basic steps required to create a web page are beyond its scope.

Required files

In order to use FormMail, the following files must exist:

- `/home/<username>/.FormMail.conf` (properly [configured](#)).
- a web page including a properly structured and configured form element.

Chapter 2: Configuration

Overview

In order for Formmail to work, you must edit the Formmail configuration file, which is located here:

```
/home/<username>/.FormMail.conf.
```

Among other things, the configuration file helps to prevent spam addressed to your domain, preserving your bandwidth for legitimate users of your site. In addition, the configuration file hides your email address from spiders and email harvesters; it also gives you access to a number of shortcuts that make it easier to develop and maintain forms. You can use [File Manager](#), [SSH](#) or [FTP](#) to work with the default configuration file.

Note: If you have any trouble editing or uploading this file, please contact support for assistance.

Default configuration file

The contents of the default configuration file (`.FormMail.conf`) are shown below.

```
#### NMS Secure FormMail v2.20 2002/11/21 (Release 1.0)
####
#### *Configuration File*
#### If any values are not set properly, FormMail WILL NOT work.
####
#### Save this file in your home directory (/home/username/)
#### named '.FormMail.conf'
# Set this to '1' if you receive any errors. They will
# Be displayed to the browser in a more verbose manner.
[DEBUGGING]
0
[/DEBUGGING]
# This address will receive bounced messages if any of the emails
# cannot be delivered, and should be set to your email address.
#
[postmaster]
you@yourdomain.com
[/postmaster]
# A list of the email addresses that formmail can send
# email to. The elements of this list can be either
# simple email addresses (like 'you@your.domain') or
# domain names (like 'your.domain'). If it's a domain
# name then *any* address at the domain will be allowed.
#
# Also see NOTE below for aliases.
#
# NOTE: One address/domain per line
#
[allow_mail_to]
yourdomain.com
```

```

you@example.com
[/allow_mail_to]
# A hash for predefining a list of recipients in the
# script, and then choosing between them using the
# recipient form field, while keeping all the email
# addresses out of the HTML so that they don't get
# collected by address harvesters and sent junk email.
#
# For example, suppose you have three forms on your
# site, and you want each to submit to a different email
# address and you want to keep the addresses hidden.
#
# In the HTML form that should submit to the recipient
# 'me@mydomain.com', you would then set the recipient
# with:
#
# <input type="hidden" name="recipient" value="me" />
#
# NOTE: If an alias is set for any email address, then it is
# not required to be in the [allow_mail_to] block, it
# is automatically allowed.
#
# NOTE: One alias per line.
#
[recipient_alias]
me=>you@yourdomain.com
him=>you@yao.com,you@hotmail.com
[/recipient_alias]
# If this flag is set to 1 then an additional email
# will be sent to the person who submitted the
# form.
#
# CAUTION: with this feature turned on it's
# possible for someone to put someone else's email
# address in the form and submit it 5000 times,
# causing this script to send a flood of email to a
# third party. This third party is likely to blame
# you for the email flood attack.
#
[send_confirmation_mail]
0
[/send_confirmation_mail]
# The header and body of the confirmation email
# sent to the person who submits the form, if the
# [send_confirmation_mail] flag is set. In the
# example below, everything between the lines:
#
#   [confirmation_text]
# and
#   [/confirmation_text]
#
# is treated as part of the email.

# !!IMPORTANT!!
# Everything before the first blank line is taken as part of
# the email header, and everything after the first
# blank line is the body of the email.
[confirmation_text]
From: you@yourdomain.com

```

```
Subject: Your Form Submission
Thank you for your submission.
[/confirmation_text]
# The Cascading Style Sheet (CSS) used for the 'thank you' page
# if a redirect is not used. This is an absolute URL.
#
# i.e. /css/site.css would be http://yourdomain.com/css/site.css
#
# This may be left blank.
#
[style]
css/site.css
[/style]
# The Character set used for parsing form data and for the resulting
# 'Thank You' page after form submission.
#
# This may be left blank.
#
[charset]
iso-8859-1
[/charset]
```

Using File Manager to edit the configuration file

- 1 Go to your control panel, and click the File Manager icon.
- 2 When the File Manager screen loads, type `.FormMail.conf` in the text area to the left of Create/Edit file, then click Create/Edit file.
- 3 Select the [provided text](#) of the configuration file with your mouse, right-click, and choose copy.
- 4 When the page loads, right-click your mouse and select paste. (If the file is not empty, you already have a configuration file, and can skip this step.)
- 5 Scroll through the text of the configuration file, and update the values to suit your needs. At the very least, you will need to enter your domain name (without the `www.`) in between the `allow_mail_to` lines, and put your email address in between the `post-master` lines.
- 6 Once your changes have been made, click Save.

Now, you can send email via forms in your html files to any email address set up on your domain. If you would like to send email to other addresses other than ones set up on your domain, simply add the email addresses in between the `allow_mail_to` lines of your configuration file.

There is more detailed information explaining this in the actual configuration file. Read over it and if you have any questions, please contact support for assistance.)

Note: Although you will not be able to see your configuration file in the File Manager, it is there. Because it begins with a 'dot' (.), it is considered hidden. You can always access it simply by typing the name in the text area manually. This configuration file contains more detailed information concerning `allow_mail_to`.

Using SSH to edit the configuration file

This method is more advanced than the File Manager. It uses a unix command prompt and vi to create the `.FormMail.conf` file. If you have any problems with this method, please try the above method, or contact support for assistance.

1 Enable SSH to your domain.

2 From the command prompt, type the following (each command is followed by a hard return):

- `cd /home/user` (where user is your domain's username)
- `vi .FormMail.conf`

3 If the file already exists, continue to the next step, otherwise, press `i` to activate insert mode, then paste the configuration file located at the end of this guide. Press `ESC` then type `:1` to go to the first line of the file.

4 Once at the top of the file, proceed to read the descriptions and become familiar with their purposes. When you come to `postmaster` and `allow_mail_to`, you will have to enter the values in between the two lines marking the boundaries of those values. The rest is optional, but very handy.

The following are vi commands you may find useful:

- Type `i` to switch to insert (editing) mode.
- Type `o` to switch insert mode at the next line below your cursor.
- Type `u` to undo your last change. Note that this command cannot be used while in insert mode.
- Press `ESC` to exit insert mode.
- Type `:w` to save your file.
- Type `:wq` to save and exit vi.
- To save your entries and exit vi, do the following: Hit the `ESC` key, type `:wq`, then hit `Enter`.

The email addresses in the `.FormMail.conf` `allow_mail_to` section should match the email addresses in the value field (e.g):

```
<input type="hidden"
name="recipient" value="username@example.com,
anothername@example.com" />
```

Your form's action line should be: `<form action="http://domainname.ext/cgi-sys/formmail.pl" method="post">`, where `domainname.ext` is your domain name.

The above is also true if you would like to use your own secure apache certificate for your forms. The only change is the `form action` line. The `http://` gets replaced with `https://`. For example, the form action line for a domain specific certificate should be:

```
<form action="https://domainname.ext/cgi-sys/formmail.pl"
method="post">
```

Uploading the configuration file via FTP

- 1 Create FormMail.conf (note the lack of a leading "dot") on your own computer with a program such as NotePad (Windows), TextEdit (Mac), or any other text-editor.
- 2 Copy the contents of .FormMail.conf [as provided](#), and make any necessary changes.
- 3 Save the file as plain-text.
- 4 Upload the file to your home directory and rename it to .FormMail.conf.

Chapter 3: Form Structure

Overview

In order to actually use FormMail, you must insert an HTML form element in the source code of the web page on which you want the form to appear. Every web-based form includes at least the following elements: tags to open and close the form, input fields to allow the user to submit information, and a button to actually submit the form.

The form itself

Your entire form must be inclosed in `<form>` tags.

```
<form>
. . . actual form content goes here . . .
</form>
```

Once you have created a frame for your form, you must determine what will be done with it once it has been submitted. This is done with `action` and `method` attributes. The `action` attribute points your browser to a file that contains processing instructions, and the `method` attribute determines how your browser will interact with that file.

Formmail requires the `action` to point to `/cgi-bin/formmail.pl` and that the `method` be `post`, so your form will actually look like this, if located in your `www` directory.:

```
<form action="/cgi-sys/formmail.pl" method="post">
. . . actual form content goes here . . .
</form>
```

Getting information from users

Form input is contained in one of several types of tags. We will discuss `input` and `textarea`, but many more are available (for more information, check out the w3c specification at <http://www.w3.org/TR/html4/interact/forms.html>).

input

The vast majority of the information you collect from users will be through `input`.

hidden Hidden inputs contain information that you want to be included in the form, but are not to be entered by the user. You may want to specify the recipient (yourself) in a hidden field.

```
<input type="hidden" name="recipient" value="youremail@yourdo-
main.ext" />
```

text Text inputs create single line text boxes into which your users can enter information, such as their email address.

```
<input type="text" name="email" />
```

radio Radio buttons must have the same name. Only one can be selected at once.

```
<input type="radio" name="color" value="red"/>Red<br />
<input type="radio" name="color" value="blue"/>Blue<br />
<input type="radio" name="color" value="yellow"/>Yellow<br />
```

textarea

If you are going to request that your users submit large blocks of text, you may want to include a larger area for them to type, instead of a small input box. The `textarea` tag requires attributes that determine its length (`cols`) and its height (`rows`). The code that follows will give you a text box that is 5 lines tall and 80 characters wide, and contains the words “Type your information here,” which the user may replace.

```
<textarea rows="5" cols="80">Type your information here</textarea>
```

Submitting the form

Users will submit your form by clicking a submit button. The HTML for this follows:

```
<button type="submit" name="submit">Submit This Form</button>
```

Sample Form

```
<form action="/cgi-sys/formmail.pl" method="post">

  <!-- ***** hidden fields ***** -->
  <!-- determines where form will be emailed -->
  <input type="hidden" name="recipient" value="user@example.com" />
  <!-- determines subject of form email -->
  <input type="hidden" name="subject" value="Email Form Example" />
  <!-- includes information about the user's browser in the form email -
  ->
  <input type="hidden" name="env_report" value="HTTP_USER_AGENT" />

  <!-- ***** fields that will appear for the user to fill out
  ***** -->
  <!-- allows the user to input his or her name -->
  Name: <br />
  <input type="text" name="realname" id="name" /><br />

  <!-- allows the user to input his or her own email address -->
  Email: <br />
```

```

<input type="text" name="email" id="email" /><br />

<!-- displays a textbox for the user to enter comments -->
Comments: <br />
<textarea name="comments" id="comments" rows="5" cols="30"></tex-
tarea><br clear="all" />

<!-- radio buttons allowing the user to choose his or her favorite
color -->
<label for="color">What is your favorite color?</label><br />
<input type="radio" name="color" value="red" id="color" />Red<br />
<input type="radio" name="color" value="blue" id="color" />Blue<br />
<input type="radio" name="color" value="yellow" id="color" />Yel-
low<br />

<!-- ***** buttons ***** -->
<button type="submit" name="submit">Submit This Form</button>

</form>

```

The above HTML, included in an HTML document, will produce a form that looks similar to the following:

Name:

Email:

Comments:

What is your favorite color?
 Red
 Blue
 Yellow

Chapter 4: Input Fields

Overview

Your form can contain as many custom input fields as you wish. The `name` attribute you set on your input field will be used to label what the user entered when the submission is sent to the [recipient](#). FormMail also recognizes special input fields to force behaviors from client software. To harness these capabilities, the fields must be named according to the designations in the following topics.

recipient

The `recipient` field designates the receiver of the email. It is the one field that must appear in your form in order for it to be processed correctly upon submission.

Field syntax

```
<input type="hidden" name="recipient" value="name@example.com,othername@example.com" />
```

Note: Email addresses included in this field must also be included in `.FormMail.conf`.

subject

The `subject` field allows you to specify the text that will appear in the subject line of the email you receive once the form is submitted. If you do not specify a subject, the default text—*WWW Form Submission*—will be used.

Field syntax

```
<input type="hidden" name="subject" value="Your Subject" />
```

Note: If you want the person submitting the form to be able to specify the subject, set the `type` attribute to `text` (instead of `hidden`).

email

When present, the `email` field allows the user to specify his or her return address. If you want to be able to respond to form-submitted emails, you should include this field.

Field syntax

```
<input type="text" name="email" />
```

realname

The `realname` field allows the user to input his or her name. This field is useful for identification purposes and is included the `From` line of the message header.

Field syntax

```
<input type="text" name="realname" />
```

sort

This field allows you to choose the order in which you wish your variables to appear in the email that FormMail generates. You can choose to have the field sorted alphabetically or specify a set order that the fields will appear in mail messages. By leaving this field out, the order will simply default to the order in which the browsers send the information to the script (which isn't always the same order the variables appear in the form.) When sorting by a set order of fields, you should include the phrase `order` as the first part of your value for the sort field, then follow with the field names you want to be listed in the email message, separated by commas.

Field syntax

To sort alphabetically:

```
<input type="hidden" name="sort" value="alphabetic" />.
```

To sort by a set field order:

```
<input type="hidden" name="sort" value="order:name1,name2,etc" />
```

redirect

Upon form submission, you may redirect the user to a different URL, rather than the default form response.

Field syntax

```
<input type="hidden" name="redirect" value="http://domain-name.ext/yourfile.html" />
```

required

You can require that users fill in certain fields before the user can successfully submit the form. Simply place all field names that you want to be mandatory into this field. If the required fields are not filled in, the user will be notified that information is missing, and a link back to the form they just submitted will be provided.

Field syntax

If you want to require that the user fill in the email and phone fields in your form, so that you can reach them once you have received the mail:

```
<input type="hidden" name="required" value="email,phone" />
```

env_report

You may include environment variables in the email message you receive after a user has filled out your form. This is useful if you wish to know what browser they were using, what

domain they were coming from, or any other attributes associated with environment variables. The following is a short list of environment variables you may find useful:

- REMOTE_HOST sends the hostname making a request.
- REMOTE_ADDR sends the IP address of the remote host making the request.
- HTTP_USER_AGENT is the browser the client is using to send the request.

Field syntax

To show remote host and browser sending the request:

```
<input type="hidden" name="env_report"
value="REMOTE_HOST,HTTP_USER_AGENT" />
```

title

This field allows you to specify the title of the form response page if you do not specify a redirect URL.

Field syntax

If you wanted the title to be Feedback Form Results:

```
<input type="hidden" name="title" value="Feedback Form Results" />
```

return_link_url

This field allows you to specify the URL that will appear on the report page if you have not set the `redirect` field.. It is useful if you allow the user to receive the form response after submission, but still want to provide link back to your main page. You may set the text of the link in the `return_link_title` field.

Field syntax

```
<input type="hidden" name="return_link_url" value="http://
your.example.com/main.html" />
```

return_link_title

This is the title that will be used to link the user back to the page you specify with `return_link_url`. The two fields will be shown on the resulting form page as:

```
<ul><li><a href="return_link_url">return_link_title</a></li></ul>
```

Field syntax

```
<input type="hidden" name="return_link_title" value="Back to Main
Page" />
```

background

This field allows you to specify the background image of your form results page, providing you have not set the `redirect` field.

Field syntax

```
<input type="hidden" name="background" value="http://domain-name.ext/yourimage.gif" />
```

bgcolor

This form field allows you to specify a background color for the form results page in much the way you specify a background image. The value must be hexadecimal, that is #RRBGGG.

Field syntax

For a white background:

```
<input type="hidden" name="bgcolor" value="#FFFFFF" />
```

text_color

This field works in the same way as `bgcolor`, except that it will change the color of your text.

Field syntax

For black text:

```
<input type="hidden" name="text_color" value="#000000" />
```

link_color

This field changes the color of links on the result page and works in the same way as `text_color`. If `redirect` is defined, then this link will have no effect.

Field syntax

For red links:

```
<input type="hidden" name="link_color" value="#FF0000" />
```

vlink_color

This field changes the color of visited links on the result page and works in the same way as `text_color`. If `redirect` is defined, then this link will have no effect.

Field syntax

For blue visited links:

```
<input type="hidden" name="vlink_color" value="#0000FF" />
```

alink_color

This field changes the color of active links on the result page and works in the same way as `text_color`. If `redirect` is defined, then this link will have no effect.

Field syntax

For a blue active links:

```
<input type="hidden" name="alink_color" value="#0000FF" />
```