

Using nmsFormail

nmsFormail is free software written in Perl, which provides the ability to send emails using data entered via website forms . In particular, it is often used for generating emails via Contact forms. It can be found at <http://nms-cgi.sourceforge.net/> (the nmsFormail title in the Miscellaneous section is a link to this site.)

NOTE: *For reasons of security the perl script must be renamed before use. This is to try and prevent hacker/spammer abusing the script.*

You will need to alter the perl script to make it work for your needs. This is actually quite easy, so don't worry.

DOWNLOAD

After you have downloaded the zip file, decompress the file and examine the contents of the folder.

NOTE: *Use a suitable program to decompress the zip file, such as WinZip (<http://www.winzip.com/>)*

You should see the following files

FormMail.pl - this is the FormMail script written in Perl, it needs to be renamed before it can be used.

README - this provides full instructions. You can open and view this file using Notepad, it is text document.

EXAMPLES - this file contains examples of using formail, this will help you understand the instructions given in this document and the README file. You can open and view this file using Notepad, it is text document.

ChangeLog - this file contains details of changes that have occurred and is useful to developers and advanced users. You can open and view this file using Notepad, it is text document.

MANIFEST - this is a list of the files. You can open and view this file using Notepad, it is text document.

AMENDMENTS TO FORMMAIL.PL

For more comprehensive instructions read the README file supplied with the FormMail download. The following discussion will configure the FormMail for a simple case.

First of all, make a copy of the script called FormMail.pl.

Second rename the copy to anything you consider suitable, but **don't use** 'formail' in the name!

REMEMBER: Work on your renamed script, though I will still be referring to it as FormMail.

You need to open the FormMail.pl script to make the changes (remember you are working with your copy of the script). If you don't have a suitable perl editor, then use Notepad in windows. This is one of many text editors, that don't add extra 'invisible' code. A text editor is **not** the same as a Word Processor such as Microsoft Word, these types of program add additional code to the document which will prevent it working.

Script

Open and have a look at the script file (FormMail.pl). It is broken into various sections.

The first part contains some comments (those with the #) and a list of the variables that are to be used in the script.

```
#!/usr/bin/perl -wT
#
# NMS FormMail Version 3.14c1
#
use strict;
use vars qw(
    $DEBUGGING $emulate_matts_code $secure %more_config
    $allow_empty_ref $max_recipients $mailprog @referers
    @allow_mail_to @recipients %recipient_alias
    @valid_ENV $date_fmt $style $send_confirmation_mail
    $confirmation_text $locale $charset $no_content
    $double_spacing $wrap_text $wrap_style $postmaster
    $address_style
);
```

Scroll past this lot.

The next bit is some comments about the program.

```
# PROGRAM INFORMATION
# -----
# FormMail.pl Version 3.14c1
#
# This program is licensed in the same way as Perl
# itself. You are free to choose between the GNU Public
# License <http://www.gnu.org/licenses/gpl.html> or
# the Artistic License
# <http://www.perl.com/pub/a/language/misc/Artistic.html>
#
# For help on configuration or installation see the
# README file or the POD documentation at the end of
# this file.
```

Next is the User Information section. This is the bit you will need to change.

```
# USER CONFIGURATION SECTION
# -----
# Modify these to your own settings. You might have to
# contact your system administrator if you do not run
# your own web server. If the purpose of these
# parameters seems unclear, please see the README file.
#
BEGIN
{
  $DEBUGGING          = 1;
  $emulate_matts_code = 0;
  $secure             = 1;
  $allow_empty_ref    = 1;
  $max_recipients     = 5;
  $mailprog           = '/usr/lib/sendmail -oi -t';
  $postmaster         = '';
  @referers           = qw(dave.org.uk 209.207.222.64 localhost);
  @allow_mail_to      = qw(you@your.domain
some.one.else@your.domain localhost);
  @recipients         = ();
  %recipient_alias    = ();
  @valid_ENV          = qw(REMOTE_HOST REMOTE_ADDR REMOTE_USER
HTTP_USER_AGENT);
  $locale             = '';
  $charset            = 'iso-8859-1';
  $date_fmt           = '%A, %B %d, %Y at %H:%M:%S';
  $style              = '/css/nms.css';
  $no_content         = 0;
  $double_spacing     = 1;
  $wrap_text          = 0;
  $wrap_style         = 1;
  $address_style      = 0;
  $send_confirmation_mail = 0;
  $confirmation_text  = <<'END_OF_CONFIRMATION';
From: you@your.com
Subject: form submission

Thank you for your form submission.

END_OF_CONFIRMATION
```

For a full explanation see the README file. We will cover just the basics.

By default \$DEBUGGER is set to 1, which is a security risk, this needs to be changed to '0' (zero) once you upload this script and make it 'live' on you site. At the moment leave it, as this setting will display any errors in your browser.

The \$max_recipients variable, as the name suggests determines the maximum number of recipients than an email can be sent to. The default is 5, but as we are only planning on receiving one email per user from the comments page, change this to 1.

The \$mailprog variable determines the path to the sendmail binary that allows the sending of emails on the hosting server. This information is provided in cPanel and should be

```
/usr/sbin/sendmail
```

Alter the \$mailprog variable to the path to sendmail as indicated in your cPanel. Leave the switches (-oi -t) as is, only change the path.

```
eg $mailprog      = '/usr/sbin/sendmail -oi -t';
```

The \$postmaster variable is contained between two single quotes (it is not a double quote on its own!). By default these are empty (""). The address that you wish to send any error should be added here; typically this would be your email address.

```
e.g. $postmaster  = 'yourdom@yourdomain.com';
```

The @referers variable is a list of the forms that can submit to this script. It should contain the url of your website and the word localhost.

```
e.g. @referers    = qw(www.yourdomain.com localhost);
```

The @allow_mail_to variable is a list of emails that the script will send email to i.e. the email you want to receive from this script and this will be your contact email address.

```
e.g. @allow_mail_to = qw(you@yourdomain.com localhost);
```

Leave the @recipients and the remaining variables as their default values. Read the README file for more information about these variables. The remaining part of the document is the Perl script and should not be amended.

Your configuration of the script should like the following:

```
$DEBUGGING          = 0;
$emulate_matts_code= 0;
$secure             = 1;
$allow_empty_ref    = 1;
$max_recipients     = 1;
$mailprog           = '/usr/sbin/sendmail -oi -t';
$postmaster         = 'yourdom@yourdomain.com';
@referers           = qw(www.yourdomain.com localhost);
@allow_mail_to     = qw(you@your.domain
some.one.else@your.domain localhost);
@recipients         = ();
%recipient_alias    = ();
@valid_ENV          = qw(REMOTE_HOST REMOTE_ADDR REMOTE_USER
HTTP_USER_AGENT);
$locale            = '';
$charset           = 'iso-8859-1';
$date_fmt          = '%A, %B %d, %Y at %H:%M:%S';
$style             = '/css/nms.css';
$no_content        = 0;
$double_spacing    = 1;
$wrap_text         = 0;
$wrap_style        = 1;
$address_style     = 0;
$send_confirmation_mail = 0;
$confirmation_text = <<'END_OF_CONFIRMATION';
```

UPLOADING THE SCRIPT

The script should be uploaded to the cgi-bin folder. This is located in the \public_html folder where all your web files should be. Use a suitable FTP client to upload the file.

REMEMBER: *The script to upload is the amended script with the new filename. You may also need to change the extension from .pl to .cgi if it does not work.*

Once the file is uploaded, change the permissions on the file to enable the file to be executed, see 'Uploading Files using FTP' in the How to section of the website and also see the tutorial section for information about file permissions.

CREATING A CONTACT FORM

To use the script you need to have a form that accesses and uses the script.

(NOTE: *FormMail, in this example has been renamed as feedback.cgi*)

```
<form method="post" action="http://www.yourdomain.com/cgi-bin/feedback.cgi">
  <input type="hidden" name="redirect"
  value="http://www.your.domain/thankyou.html" />
  <input type="text" name="feedback" /><br />
  Please enter your comments<br />
  <input type="submit" />
</form>
```

Notice the hidden input in the form. This allows the user to be redirected to a thank you page after they have submitted the feedback form.

Create a HTML page called thankyou.html that thanks the user for completing the form.

There are a number of other useful hidden and unhidden inputs that are covered in the README file.

Upload your contact page and thankyou page to the web server.

TESTING

Now you have your HTML pages and the FormMail script uploaded to the server, you can test them together.

Complete the contact form and click the Submit button, the form input values are now POSTed to the script and you should be redirected to the 'thank you' page. The email you set up in the script should receive details that were completed in the form.

Errors

Any errors should appear in your browser. Check through the script carefully for mistakes if you see an error. Errors often occur because you have accidentally overwritten a semicolon or quote mark. At the end of the README file is a list of common mistakes.

If every thing is working correctly, re-open the script and change the \$DEBUGGING variable to 0 and upload to the server.

You may what to start adding more input to the contact form, such a name , address, email etc. You can now experiment to develop the form from this basic introduction.