

# How To Use Form Mail

FormMail is a generic www form to e-mail gateway that will parse the results of any form and send them to the specified user. It is a program that allows you to send an e-mail from an online form and can be downloaded from <http://www.scriptarchive.com/formmail.html> or from <http://www.scriptarchive.com/nms.html> if you prefer the nms version. This script has many formatting and operational options, most of which can be specified through the form, so you don't need any programming knowledge or multiple scripts for multiple forms.

- [Fields](#)
  - [Installing FormMail](#)
  - [Using FormMail](#)
  - [Recipient Field](#)
  - [Subject Field](#)
  - [E-mail Field](#)
  - [Realname Field](#)
  - [Redirect Field](#)
  - [Required Field](#)
  - [Env\\_report Field](#)
  - [Sort Field](#)
  - [Print\\_config Field](#)
  - [Print\\_blank\\_fields Field](#)
  - [Title Field](#)
  - [Return\\_link\\_url Field](#)
- [Advanced FormMail Script Configuration](#)
  - [VALID:](#)
  - [INVALID:](#)

## Fields

### Installing FormMail

Please visit <http://www.scriptarchive.com/download.cgi?s=formmail&c=txt&f=README> for the installation instructions for installing this script. You may also choose to download the nms version at <http://www.scriptarchive.com/nms-download.cgi?s=nms-formmail&c=zip> which Replaces: FormMail v1.92 @ MSA if you choose. This knowledge base article will be dealing with the formmail script and not the nms version.

### Using FormMail

In order for an e-mail to be delivered to an e-mail on the local server, the e-mail account must have a home directory. The home directory can be created by enabling ftp, it can then be disabled if desired, but needs to have been enabled once or the FormMail program will not find the e-mail account (the primary account will not need this step).

You need to create the page for your form. Add the code that is listed below to you page. You only need the recipient form field in your form for FormMail to work correctly. Other hidden configuration fields can be used to enhance the operation of FormMail on your site. The action of your form needs to point towards this script, and the method must be POST in capital letters. Here is an example of the form fields to put in your form:

```
< FORM METHOD=POST ACTION="http://yourdomain.com/cgi-bin/formmail/FormMail.pl">
< input type=hidden name="recipient" value="whoever@yourdomain.com">
< input type=hidden name="subject" value="Order">
< input type=hidden name="return_link_url" value="http://yourdomain.com/">
< input type=hidden name="return_link_title" value="Back to Main Page">
```

The following are descriptions and proper syntax for fields you can use with FormMail.

### Recipient Field

Description: This form field allows you to specify to whom you want your form results mailed to. You will likely want to configure this option as a hidden form field with a value equal to that of your e-mail address.

Syntax: `< input type=hidden name="recipient" value="email@yourdomain.com">`

### Subject Field

Description: The subject field will allow you to specify the subject that you want to appear in the e-mail that is sent to you after this form has been filled out. If you do not have this option turned on, then the script will default to a message subject "WWW Form Submission."

Syntax: If you want to choose what the subject is:

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

To allow the user to choose a subject:

```
< input type=text name="subject">
```

### E-mail Field

Description: This form field will allow users to specify their return e-mail address. If you want to be able to return e-mail to your user, we strongly suggest that you include this form field. This will be put into the From: field of the message you receive. If you want to require an e-mail address with valid syntax, add this field name to the 'required' field.

Syntax: < input type=text name="email">

## Realname Field

Description: The realname form field will allow users to insert their real name. This field is useful for identification purposes and will also be put into the From: line of your message header.

Syntax: < input type=text name="realname">

## Redirect Field

Description: If you want to redirect the user to a different URL rather than having them see the default response to the fill-out form, you can use this hidden variable to send them to a pre-made HTML page.

Syntax: To choose the URL they will end up at:

```
< input type=hidden name="redirect" value="http://yourdomain.com/to/file.html">
```

To allow them to specify a URL they want to travel to once the form is filled out:

```
< input type=text name="redirect">
```

## Required Field

Description: You can require that certain fields in your form are filled in before the user can successfully submit the form. Simply place all field names that you want to be mandatory into this field, separated by commas. If the required fields are not filled in, the user will be notified of what they need to fill in, and a link back to the form they just submitted will be provided. To use a customized error page, see 'missing\_fields\_redirect'

Syntax: If you want to require that they fill in the e-mail and phone fields in your form so that you can reach them once you have received the e-mail, use the syntax like:

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

## Env\_report Field

Description: This allows you to have environment variables included in the e-mail message you receive after a user has filled out your form. This is useful if you want 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 valid environment variables that might be useful:

REMOTE\_HOST: Sends the hostname making the request.

REMOTE\_ADDR: Sends the IP address of the remote host.

HTTP\_USER\_AGENT: The browser the client is using.

NOTE: In our case, both REMOTE\_HOST and REMOTE\_ADDR are the same since our servers don't do the reverse DNS lookup needed to generate the true REMOTE\_HOST string.

Syntax: For all the above variables, put the following into your form:

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

## Sort Field

Description: This field allows you to choose the order in which you want your variables to appear in the e-mail form that FormMail generates. You can choose to have the field sorted alphabetically or specify a set order in which you want the fields to appear in your mail message. By leaving this field out, the order will simply default to the order in which the browsers send the information to the script, which is usually the exact same order as they appeared 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, and then follow that with the field names you want to be listed in the e-mail message, separated by commas.

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...">
```

## Print\_config Field

Description: print\_config allows you to specify which of the config variables you would like to have printed in your e-mail message. By default, no config fields are printed to your e-mail. This is because the important form fields, like e-mail, subject, etc. are included in the header of the message. However, some users have asked for this option so they can have these fields printed in the body of the message. The config fields that you want to have printed should be in the value attribute of your input tag separated by commas.

Syntax: If you want to print the e-mail and subject fields in the body of your message, you would place the following form tag:

```
< input type=hidden name="print config" value="email, subject">
```

## Print\_blank\_fields Field

Description: print\_blank\_fields allows you to request that all form fields are printed in the return HTML, regardless of whether or not they were filled in.

FormMail defaults to turning this off so that unused form fields aren't e-mailed.

Syntax : < input type=hidden name="print\_blank\_fields" value="1">

## Title Field

Description: This form field allows you to specify the title and header that will appear on the resulting page if you do not specify a redirect URL.

Syntax: If you wanted a title of 'Feedback Form Results:'

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

## **Return\_link\_url Field**

Description: This field allows you to specify a URL that will appear as return\_link\_title on the following report page. This field will not be used if you have the redirect field set, but it is useful if you allow the user to receive the report on the following page but want to offer them a way to get back to your main page.

Syntax: < input type=hidden name="return\_link\_url" value="http://yourdomain.com/index.htm">

Return\_link\_title Field

Description: 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:

Back to Main Page

Syntax: < input type=hidden name="return\_link\_title" value="Back to Main Page">

## **Advanced FormMail Script Configuration**

### **Setting up the FormMail Script**

The [FormMail.pl](#) script does not need to be configured extensively in order for it to work. There are only two variables in the perl file that you will need to define along with changing the top line of your script to match the location of your Perl interpreter.

```
$mailprog = '/usr/lib/sendmail -i -t';
```

This variable must define the location to your server's sendmail program. If this is incorrect, form results will not be mailed to you. Specifying the parameters in this variable is new in v1.91, and we have included the -i parameter so that a single period on a line by itself will not end the message. -t instructs sendmail to read the recipient list from the message text.

```
@referers = ('scriptarchive.com',YOUR_IP');
```

This array allows you to define the domains on which you allow forms to reside and use this installation of FormMail. If a user tries to put a form on another server that is not [scriptarchive.com](#), they will receive an error message when someone tries to fill out their form. By placing [scriptarchive.com](#) in the @referers array, this also allows [www.scriptarchive.com](#), [ftp.scriptarchive.com](#), any other http address with [scriptarchive.com](#) in it, and [scriptarchive.com](#)'s IP address to access this script as well, so no users will be turned away.

NOTE: This is not a security check. Referer headers can EASILY be faked. Rather, it prevents someone on [xyznotyou.com](#) from using the FormMail on your server to process forms on their server on a regular basis. It remains in the script as a remnant of earlier versions when it was used for security, but the @recipients variable is now used to specify exactly who can receive e-mail from this installation.

As of version 1.7, the domains listed here are also used as the defaults when checking valid recipient e-mail addresses. You should either include all domain names that you want to have FormMail send e-mails to in your @referers array or tailor the @recipients array by hand.

```
@valid_ENV = ('REMOTE_HOST','REMOTE_ADDR','REMOTE_USER','HTTP_USER_AGENT');
```

This array allows the administrator to specify a list of environment variables that the user may request be added into the e-mail. This is a security patch that was advised at <http://www.securityfocus.com/bid/1187> and was implemented by Peter D. Thompson Yezek at <http://www.securityfocus.com/archive/1/62033>.

Only environment variables listed in this array may be included in the form field env\_report, so if you wanted to also know what URL a user was submitting from, you could change @valid\_ENV to:

```
@valid_ENV = ('REMOTE_HOST','REMOTE_ADDR','REMOTE_USER',  
'HTTP_USER_AGENT','HTTP_REFERER');
```

and then include HTTP\_REFERER in your env\_report form field.

```
@recipients = &fill_recipients(@referers);
```

If you want to only allow e-mail addresses at the domain names in @referers to receive form results, you probably do not need to change this variable. However, if you get any 'Error: Bad/No Recipient' messages when running FormMail, you might have to revisit @recipients and make sure you have correctly listed all domains or configured this variable.

@recipients is the most important variable you need to configure. It is an array of regular expressions defining all valid recipients that can be specified. In order for an e-mail to be sent to the recipient defined in a form, the recipient e-mail address must match one of the elements in the @recipients array. For the most simple setup, place any domain name that you want to send form results to in the @referers array.

**WARNING:** This allows those domains to also access your FormMail script and utilize it to process their own forms, but likely this is what you intended anyway. If so, you can set the @referers to @recipients = &fill\_recipients(@referers); If not, another alternative is to set @recipients equal to the return value of the fill\_recipients function and pass this function all of the domains to which e-mail may be addressed: @recipients = &fill\_recipients('domain.com','sub.domain.com','another.com');

You are now allowing e-mail to any username (provided it contains only A-Z, a-z, 0-9, \_, - or .) at those three domains.

Similarly, since @recipients is just an array, you could even do:

```
@recipients = (&fill_recipients('domain.com','sub.domain.com'),
```

```
'^otheruser1@otherhost\.com',
```

```
'^otheruser2@otherhost\.com');
```

This would allow any recipient at domain.com and sub.domain.com similar to the previous example, but would also allow your friends otheruser1 and otheruser2 on otherhost.com to use your FormMail! Of course, you will need to add otherhost.com into your @referers array if a form is on their host!

Here's how it works. When the fill\_recipients function is called on an array of domain names, it turns them into regular expressions. These regular expressions will only allow e-mail messages to go to a recipient with an e-mail address in the following format: [A-Za-z0-9\_-\.] + @domain.com where domain.com is specified in @referers. For any IP addresses in @referers, the following address formats are valid: [A-Za-z0-9\_-\.] + @[192.168.1.1] where 192.168.1.1 is the specified IP address in @referers.

In other words, the only valid addresses are those to usernames that include only letters, numbers, underscores, dashes, or periods and an exact domain name or IP address that was specified in the @referers array. Depending on your needs, this may be too broad or not broad enough.

The way FormMail validates a recipient address is to check the supplied recipient(s) in the submitted form against each element in the array @recipients (which is a list of Perl regular expressions). If any valid recipients are found, they will receive a copy of the message.

Using the examples of @referers = ('domain.com','192.168.1.1'); and the default usage of setting @recipients = &fill\_recipients(@referers), the contents of @recipients are now the same as if you had written:

```
@recipients = ('^[w\-\.] + @domain\.com',
```

```
'^[w\-\.] + @[192\.168\.1\.1]');
```

What these regular expressions instruct FormMail to do is require that any e-mail address passed in as a recipient of the form submission match at least one of those two formats. The following are examples of valid and invalid recipients for this exact setup:

## VALID:

```
user@domain.com, First.Last@domain.com,  
Last-First@domain.com, user_name@domain.com,  
user023@domain.com, user@[192.168.1.1],  
First.Last@[192.168.1.1], user023@[192.168.1.1],  
Last-First@[192.168.1.1], user_name@[192.168.1.1], etc.
```

## INVALID:

(using these in your form field 'recipient' will trigger error)

```
user%name@domain.com, user(name)@domain.com,  
first:last@domain.com ,  
domain.com, user@192.168.1.1,  
user@newdomain.com, user@sub.domain.com,  
user@domainname.com
```

Essentially, it only allows A-Z, a-z, 0-9, \_, -, and . in the local address area (before the @, represented as [w\-.]+ in regular expression speak) and requires the domain name to match exactly. When mailing to an IP address, it must be enclosed in [].

Some people might need to match more characters than that. Let's say you need to be able to deliver e-mail to an address like: last:first@domain.com. This requires that the ':' character now be allowed into the portion of the recipient field before the domain name. You could then modify @recipients to read @recipients = ('^[w\-.:]+@domain.com');

**WARNING:** You will need to be careful. Allowing certain characters could be VERY dangerous, especially if the characters are: %, <, >, (, ) or any new lines. You can read: <http://web.nps.navy.mil/~miller/percent-hack.html> for information on exactly why the % character could be dangerous. Also, the document that prompted 1.91 explains why some of the others could lead to problems: <http://www.monkeys.com/anti-spam/formmail-advisory.pdf>

Some might want only certain addresses to work. Let's say you only want yourself@yourdomain.com to be able to receive any form submissions. You should then set the @recipients array to @recipients = ('^yourself@yourdomain.com');

Now the only valid recipient is that one e-mail address. If there are several, simply do:

```
@recipients = ('^user1@yourdomain.com',  
'^user2@their.domain.com');
```

Prior versions of FormMail recommended settings for @recipients like:

```
@recipients = ('domain.com','192.168.1.1');
```

OR

```
@recipients = ('^joe@somewhereelse.com');
```

The first is bad because it can be easily tricked by submitting a recipient such as spamvictim%elsewhere.com@domain.com. The second is MUCH better, but since it is used as a regular expression, and '.' can mean ANY character, a hacker could use joe@somewhereelseXcom to get past a valid recipient check. This is not a very big deal in most cases.

Some people wonder what the ^ and \ characters are used for. In regular expressions, the ^ means "beginning of string." By default, FormMail places a \$ at the end of the match, which means "end of string." By using both ^ and \$ in regular expression matching, FormMail can match a string exactly. You only need to worry about including the ^, which is STRONGLY recommended for all regular expressions in the array.

The \ character is used to escape a character that otherwise means something special in regular expressions. For instance, you now see every '.' being escaped with a '\', as '.' means ANY CHARACTER, whereas '\.' requires that it match ONLY a period.

FormMail is a generic www form to e-mail gateway that will parse the results of any form and send them to the specified user. It is a program that allows you to send an e-mail from an online form. This script has many formatting and operational options, most of which can be specified through the form, so you don't need any programming knowledge or multiple scripts for multiple forms.

Using FormMail

In order for an e-mail to be delivered to an e-mail on the local server, the e-mail account must have a home directory. The home directory can be created by enabling ftp, it can then be disabled if desired, but needs to have been enabled once or the FormMail program will not find the e-mail account (the primary account will not need this step).

You need to create the page for your form. Add the code that is listed below to your page. You only need the recipient form field in your form for FormMail to work correctly. Other hidden configuration fields can be used to enhance the operation of FormMail on your site. The action of your form needs to point towards this script, and the method must be POST in capital letters. Here is an example of the form fields to put in your form:

```
< FORM METHOD=POST ACTION="http://yourdomain.com/cgi-bin/formmail/FormMail.pl">  
< input type=hidden name="recipient" value="whoever@yourdomain.com">  
< input type=hidden name="subject" value="Order">  
< input type=hidden name="return_link_url" value="http://yourdomain.com/">  
< input type=hidden name="return_link_title" value="Back to Main Page">
```

The following are descriptions and proper syntax for fields you can use with FormMail.

Recipient Field

Description: This form field allows you to specify to whom you want your form results mailed to. You will likely want to configure this option as a hidden form field with a value equal to that of your e-mail address.

Syntax: < input type=hidden name="recipient" value="email@yourdomain.com">

Subject Field

Description: The subject field will allow you to specify the subject that you want to appear in the e-mail that is sent to you after this form has been filled out. If you do not have this option turned on, then the script will default to a message subject "WWW Form Submission."

Syntax: If you want to choose what the subject is:

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

To allow the user to choose a subject:

```
< input type=text name="subject">
```

E-mail Field

Description: This form field will allow users to specify their return e-mail address. If you want to be able to return e-mail to your user, we strongly suggest that you include this form field. This will be put into the From: field of the message you receive. If you want to require an e-mail address with valid syntax, add this field name to the 'required' field.

Syntax: < input type=text name="email">

Realname Field

Description: The realname form field will allow users to insert their real name. This field is useful for identification purposes and will also be put into the From: line of your message header.

Syntax: < input type=text name="realname">

Redirect Field

Description: If you want to redirect the user to a different URL rather than having them see the default response to the fill-out form, you can use this hidden variable to send them to a pre-made HTML page.

Syntax: To choose the URL they will end up at:

```
< input type=hidden name="redirect" value="http://yourdomain.com/to/file.html">
```

To allow them to specify a URL they want to travel to once the form is filled out:

```
< input type=text name="redirect">
```

#### Required Field

Description: You can require that certain fields in your form are filled in before the user can successfully submit the form. Simply place all field names that you want to be mandatory into this field, separated by commas. If the required fields are not filled in, the user will be notified of what they need to fill in, and a link back to the form they just submitted will be provided. To use a customized error page, see 'missing\_fields\_redirect'

Syntax: If you want to require that they fill in the e-mail and phone fields in your form so that you can reach them once you have received the e-mail, use the syntax like:

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

#### Env\_report Field

Description: This allows you to have environment variables included in the e-mail message you receive after a user has filled out your form. This is useful if you want 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 valid environment variables that might be useful:

REMOTE\_HOST: Sends the hostname making the request.

REMOTE\_ADDR: Sends the IP address of the remote host.

HTTP\_USER\_AGENT: The browser the client is using.

NOTE: In our case, both REMOTE\_HOST and REMOTE\_ADDR are the same since our servers don't do the reverse DNS lookup needed to generate the true REMOTE\_HOST string.

Syntax: For all the above variables, put the following into your form:

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

#### Sort Field

Description: This field allows you to choose the order in which you want your variables to appear in the e-mail form that FormMail generates. You can choose to have the field sorted alphabetically or specify a set order in which you want the fields to appear in your mail message. By leaving this field out, the order will simply default to the order in which the browsers send the information to the script, which is usually the exact same order as they appeared 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, and then follow that with the field names you want to be listed in the e-mail message, separated by commas.

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...">
```

#### Print\_config Field

Description: print\_config allows you to specify which of the config variables you would like to have printed in your e-mail message. By default, no config fields are printed to your e-mail. This is because the important form fields, like e-mail, subject, etc. are included in the header of the message. However, some users have asked for this option so they can have these fields printed in the body of the message. The config fields that you want to have printed should be in the value attribute of your input tag separated by commas.

Syntax: If you want to print the e-mail and subject fields in the body of your message, you would place the following form tag:

```
< input type=hidden name="print config" value="email, subject">
```

#### Print\_blank\_fields Field

Description: print\_blank\_fields allows you to request that all form fields are printed in the return HTML, regardless of whether or not they were filled in. FormMail defaults to turning this off so that unused form fields aren't e-mailed.

Syntax: < input type=hidden name="print\_blank\_fields" value="1">

#### Title Field

Description: This form field allows you to specify the title and header that will appear on the resulting page if you do not specify a redirect URL.

Syntax: If you wanted a title of 'Feedback Form Results:'

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

#### Return\_link\_url Field

Description: This field allows you to specify a URL that will appear as return\_link\_title on the following report page. This field will not be used if you have the redirect field set, but it is useful if you allow the user to receive the report on the following page but want to offer them a way to get back to your main page.

Syntax: < input type=hidden name="return\_link\_url" value="http://yourdomain.com/index.htm">

#### Return\_link\_title Field

Description: 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:

Back to Main Page

Syntax: < input type=hidden name="return\_link\_title" value="Back to Main Page">

#### Advanced FormMail Script Configuration

##### Setting up the FormMail Script

The [FormMail.pl](#) script does not need to be configured extensively in order for it to work. There are only two variables in the perl file that you will need to define along with changing the top line of your script to match the location of your Perl interpreter.

```
$mailprog = '/usr/lib/sendmail -i -t';
```

This variable must define the location to your server's sendmail program. If this is incorrect, form results will not be mailed to you. Specifying the parameters in this variable is new in v1.91, and we have included the -i parameter so that a single period on a line by itself will not end the message. -t instructs sendmail to read the recipient list from the message text.

```
@referers = ('scriptarchive.com','YOUR_IP');
```

This array allows you to define the domains on which you allow forms to reside and use this installation of FormMail. If a user tries to put a form on another server that is not [scriptarchive.com](http://scriptarchive.com), they will receive an error message when someone tries to fill out their form. By placing [scriptarchive.com](http://scriptarchive.com) in the @referers array, this also allows [www.scriptarchive.com](http://www.scriptarchive.com), [ftp.scriptarchive.com](http://ftp.scriptarchive.com), any other http address with [scriptarchive.com](http://scriptarchive.com) in it, and [scriptarchive.com](http://scriptarchive.com)'s IP address to access this script as well, so no users will be turned away.

NOTE: This is not a security check. Referrer headers can EASILY be faked. Rather, it prevents someone on [xyznotyou.com](http://xyznotyou.com) from using the FormMail on your server to process forms on their server on a regular basis. It remains in the script as a remnant of earlier versions when it was used for security, but the @recipients variable is now used to specify exactly who can receive e-mail from this installation.

As of version 1.7, the domains listed here are also used as the defaults when checking valid recipient e-mail addresses. You should either include all domain names that you want to have FormMail send e-mails to in your @referers array or tailor the @recipients array by hand.

```
@valid_ENV = ('REMOTE_HOST','REMOTE_ADDR','REMOTE_USER','HTTP_USER_AGENT');
```

This array allows the administrator to specify a list of environment variables that the user may request be added into the e-mail. This is a security patch that was advised at <http://www.securityfocus.com/bid/1187> and was implemented by Peter D. Thompson Yezek at <http://www.securityfocus.com/archive/1/62033>.

Only environment variables listed in this array may be included in the form field env\_report, so if you wanted to also know what URL a user was submitting from, you could change @valid\_ENV to:

```
@valid_ENV = ('REMOTE_HOST','REMOTE_ADDR','REMOTE_USER',  
'HTTP_USER_AGENT','HTTP_REFERER');
```

and then include HTTP\_REFERER in your env\_report form field.

```
@recipients = &fill_recipients(@referers);
```

If you want to only allow e-mail addresses at the domain names in @referers to receive form results, you probably do not need to change this variable. However, if you get any 'Error: Bad/No Recipient' messages when running FormMail, you might have to revisit @recipients and make sure you have correctly listed all domains or configured this variable.

@recipients is the most important variable you need to configure. It is an array of regular expressions defining all valid recipients that can be specified. In order for an e-mail to be sent to the recipient defined in a form, the recipient e-mail address must match one of the elements in the @recipients array. For the most simple setup, place any domain name that you want to send form results to in the @referers array.

WARNING: This allows those domains to also access your FormMail script and utilize it to process their own forms, but likely this is what you intended anyway. If so, you can set the @referers to @recipients = &fill\_recipients(@referers); If not, another alternative is to set @recipients equal to the return value of the fill\_recipients function and pass this function all of the domains to which e-mail may be addressed: @recipients = &fill\_recipients('domain.com','sub.domain.com','another.com');

You are now allowing e-mail to any username (provided it contains only A-Z, a-z, 0-9, \_, - or .) at those three domains.

Similarly, since @recipients is just an array, you could even do:

```
@recipients = (&fill_recipients('domain.com','sub.domain.com'),
```

```
 '^otheruser1@otherhost\.com',
```

```
 '^otheruser2@otherhost\.com');
```

This would allow any recipient at domain.com and sub.domain.com similar to the previous example, but would also allow your friends otheruser1 and otheruser2 on otherhost.com to use your FormMail! Of course, you will need to add otherhost.com into your @referers array if a form is on their host! Here's how it works. When the fill\_recipients function is called on an array of domain names, it turns them into regular expressions. These regular expressions will only allow e-mail messages to go to a recipient with an e-mail address in the following format: [A-Za-z0-9\_-\.]@domain.com where domain.com is specified in @referers. For any IP addresses in @referers, the following address formats are valid: [A-Za-z0-9\_-\.]@[192.168.1.1] where 192.168.1.1 is the specified IP address in @referers.

In other words, the only valid addresses are those to usernames that include only letters, numbers, underscores, dashes, or periods and an exact domain name or IP address that was specified in the @referers array. Depending on your needs, this may be too broad or not broad enough.

The way FormMail validates a recipient address is to check the supplied recipient(s) in the submitted form against each element in the array @recipients (which is a list of Perl regular expressions). If any valid recipients are found, they will receive a copy of the message.

Using the examples of @referers = ('domain.com','192.168.1.1'); and the default usage of setting @recipients = &fill\_recipients(@referers), the contents of @recipients are now the same as if you had written:

```
@recipients = ('^[w\-\.\+]\@domain\.com',
```

```
 '^[w\-\.\+]\@[192\.168\.1\.1]');
```

What these regular expressions instruct FormMail to do require that any e-mail address passed in as a recipient of the form submission match at least one of those two formats. The following are examples of valid and invalid recipients for this exact setup:

VALID:

user@domain.com, First.Last@domain.com,  
Last-First@domain.com, user\_name@domain.com,  
user023@domain.com, user@[192.168.1.1],  
First.Last@[192.168.1.1], user023@[192.168.1.1],  
Last-First@[192.168.1.1], user\_name@[192.168.1.1], etc.

INVALID: (using these in your form field 'recipient' will trigger error)

user%name@domain.com, user(name)@domain.com,  
first:last@domain.com ,  
domain.com, user@192.168.1.1,  
user@newdomain.com, user@sub.domain.com,  
user@domainname.com

Essentially, it only allows A-Z, a-z, 0-9, \_, -, and . in the local address area (before the @, represented as [w\-\.] in regular expression speak) and requires the domain name to match exactly. When mailing to an IP address, it must be enclosed in [].

Some people might need to match more characters than that. Let's say you need to be able to deliver e-mail to an address like: last:first@domain.com.

This requires that the ':' character now be allowed into the portion of the recipient field before the domain name. You could then modify @recipients to read  
@recipients = ('^[w\-\.:]+\@domain\.com');

WARNING: You will need to be careful. Allowing certain characters could be VERY dangerous, especially if the characters are: %, <, >, (, ) or any new lines. You can read: <http://web.nps.navy.mil/~miller/percent-hack.html> for information on exactly why the % character could be dangerous. Also, the document that prompted 1.91 explains why some of the others could lead to problems: <http://www.monkeys.com/anti-spam/formmail-advisory.pdf>

Some might want only certain addresses to work. Let's say you only want yourself@yourdomain.com to be able to receive any form submissions. You should then set the @recipients array to @recipients = ('^yourself\@yourdomain\.com');

Now the only valid recipient is that one e-mail address. If there are several, simply do:

```
@recipients = ('^user1\@yourdomain\.com',
```

```
^user2\@their\.domain\.com');
```

Prior versions of FormMail recommended settings for @recipients like:

```
@recipients = ('domain.com','192.168.1.1');
```

OR

```
@recipients = ('^joe@somewhereelse.com');
```

The first is bad because it can be easily tricked by submitting a recipient such as spamvictim%elsewhere.com@domain.com. The second is MUCH better, but since it is used as a regular expression, and '.' can mean ANY character, a hacker could use joe@somewhereelseXcom to get past a valid recipient check. This is not a very big deal in most cases.

Some people wonder what the ^ and \ characters are used for. In regular expressions, the ^ means "beginning of string." By default, FormMail places a \$ at the end of the match, which means "end of string." By using both ^ and \$ in regular expression matching, FormMail can match a string exactly. You only need to worry about including the ^, which is STRONGLY recommended for all regular expressions in the array.

The \ character is used to escape a character that otherwise means something special in regular expressions. For instance, you now see every '.' being escaped with a '\', as '.' means ANY CHARACTER, whereas '\.' requires that it match ONLY a period.