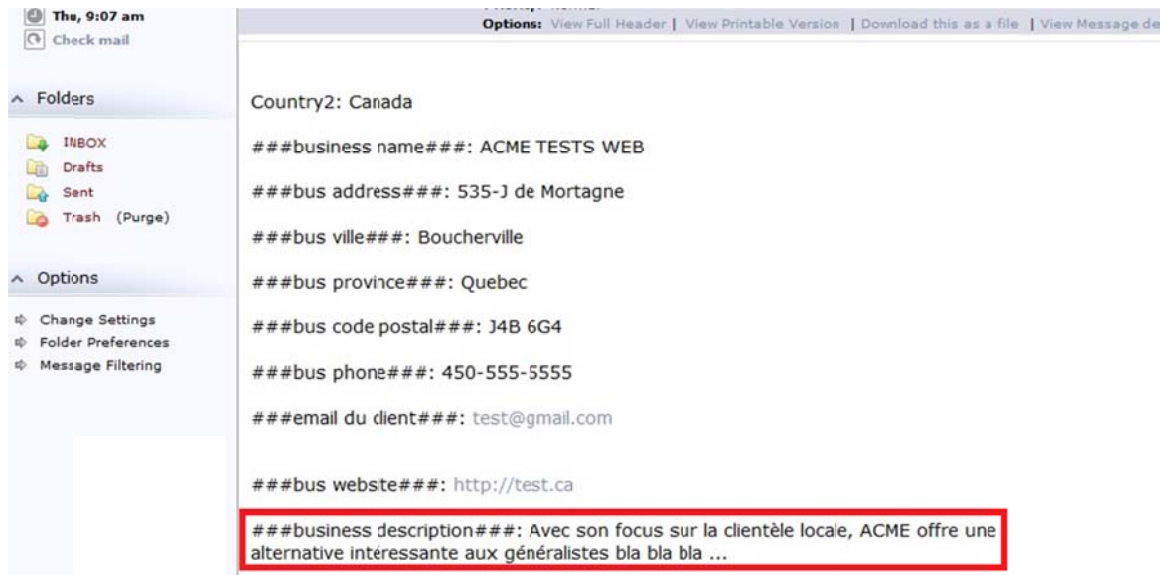


Bonjour,

J'utilise un script pour traiter un formulaire sur mon site (ci-dessous). Lorsque je lis le email envoyé dans le programme de base de mon hébergeur (webmail), tout s'affiche bien



(ci-dessous, vous trouverez l'original du email incluant le header) par contre, si je m'envoie le courriel et je l'ouvre sur mon pc (Windows 7 64 et Outlook 2010), mes accents sont transformés comme suit :

```
###business description###: Avec son focus sur la clientÃle locale, ACME offre une alternative intÃressante aux gÃnÃralistes bla bla bla ...
```

Si ce n'était que ça, je pourrais vivre avec (en le faisant reconvertir par mon script) mais lorsque j'essaie de traiter l'information avec un script, tous mes accents sont transformés en « ?? » comme ceci :

```
###business description###: avec son focus sur la client??le locale, l'Agenda Familial offre une alternative int??ressante aux bottins g??n??ralistes ..
```

Les emails que je reçois ressemblent à ceci (header):

Return-path: <bounce-cgi-nf.lution@eigbox.net >
Envelope-to: af@o.com
Delivery-date: Thu, 25 Apr 2013 08:38:07 -0400
Received: from bosimpout04.eigbox.net ([10.20.55.4])
by bosmailscan13.eigbox.net with esmtp (Exim)
id 1UVLR9-0004Y3-Ky
for af@o.com ; Thu, 25 Apr 2013 08:38:07 -0400
Received: from boscgi4404.eigbox.net ([10.20.12.127])
by bosimpout04.eigbox.net with NO UCE
id UCe8110032kU2ma01Ce87S; Thu, 25 Apr 2013 08:38:08 -0400
X-Authority-Analysis: v=2.0 cv=dt1Z+ic4 c=1 sm=1 a=ZU5MoXG1q2cA:10
a=URuuo6IAG5QA:10 a=IkCTkHD0fZMA:10 a=gjlFSQ_8AAAA:8
a=QPcu4mC3AAAA:8
a=ZZBvtSjt7_4A:10 a=pGLkceISAAAA:8 a=8gYbux20AAAA:8 a=p1MN4-
wxrEDTPLxlp8QA:9
a=QEXdDO2ut3YA:10 a=q8hKd8aIe-QA:10 a=V_bPgcFnw6kA:10
a=dzO17DpDZb0A:10
a=MSl-tDqOz04A:10 a=jIwez9dfcWwA:10 a=nNM0XHHcyIdZtYVt:21
a=2m5yBoCg1FSWoBQi:21 a=T8P1Xy7+DmnBrlhhh77XwQ==:117
X-EN-OrigOutIP: 10.20.12.127
X-EN-IMPSID: UCe8110032kU2ma01Ce87S
Received: from nf.lution by boscgi4404.eigbox.net with local (Exim)
id 1UVLQs-0005Ep-T4
for af@o.com ; Thu, 25 Apr 2013 08:37:50 -0400
X-EN-Info: U=nf.lution P=/Admin/form.php
X-EN-CGIUser: nf.lution
X-EN-CGIPath: /Admin/form.php
X-EN-OrigIP: 173.248.236.225
Message-Id: <1366893470-279-nf.inforadiosolution@boscgi4404.eigbox.net>
To: af@o.com
Subject: Google Places and Web Page
From: af@o.com
Reply-To:
MIME-Version: 1.0
Content-Type: text/plain; charset="utf-8"
X-EN-Timestamp: Thu, 25 Apr 2013 08:37:50 -0400
Date: Thu, 25 Apr 2013 08:37:50 -0400
Sender: af@o.com

(body que j'ai copié directement dans « webmail » qui est le programme de base offert par mon hébergeur pour voir les emails. Notez que les accents sont OK.

###business name###: ACME TESTS WEB

###bus address###: 535-J de Mortagne

###bus ville###: Boucherville

###bus province###: Quebec

###bus code postal###: J4B 6G4

###bus phone###: 450-555-5555

###email du client###: test@gmail.com

###bus website###: <http://test.ca>

###business description### Avec son focus sur la clientèle locale, ACME offre une alternative intéressante aux généralistes bla bla bla ...

###type d'entreprise GOOGLE###: Agence de publicité

###type d'entreprise 1###: pages jaunes

###type d'entreprise 2###: publicite locale

###type d'entreprise 3###: 411

###type d'entreprise 4###: annuaire local

###ville 1###: Drummondville

###ville 2###: Brossard

###ville 3###: Longueuil

###cellulaire du client###: 5555555555

###email###: info@mmtseo.com

###URL MOBILE VERSION###: http://

###lundil###: 09

###lundil2###: 18

###mardil###: 09

###mardi2###: 18

###mercredil###: 09

###mercredi2###: 18

###jeudi1###: 09

###jeudi2###: 21

###vendredi1###: 09

###vendredi2###: 21

###samedi1###: 09

###samedi2###: 17

###dimanche1###: Fermé

###dimanche2###: Fermé

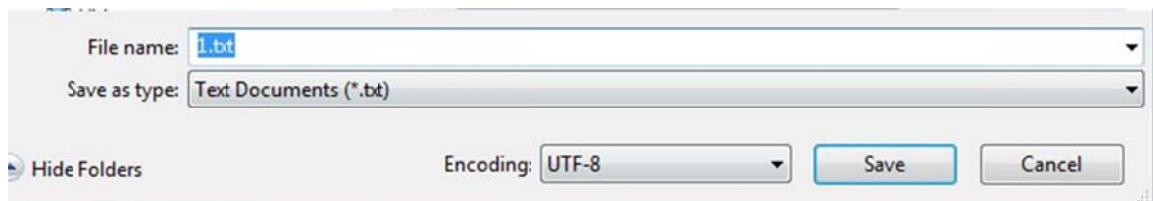
Soumettre: Soumettre

Date submitted: April 25th, 2013. 08:37 am

Ensuite, j'utilise un script pour traiter l'information hors, bien que je demande une inscription en utf-8, j'obtiens ceci:

Avec son focus sur la client??le locale, ACME offre une alternative int??ressante |..

Hors, le document est bien en utf-8 :



J'utilise ce script pour traiter un formulaire sur mon site web (j'ai mis seulement la partie configurable par respect pour l'auteur et la partie où on configure l'utf-8 est en jaune pour vous épargner du temps) :

F<?php

```
error_reporting(E_ALL ^ E_NOTICE);
```

```
/*
```

Thank you for purchasing FormToEmail-Pro by FormToEmail.com

Version 3.6 September 30th 2009

COPYRIGHT FormToEmail.com 2006 - 2009 - All rights reserved.

You may not sell (or otherwise distribute) this script. YOU ARE NOT PERMITTED TO POST THIS CODE IN A PUBLIC PLACE, such as a forum for example. A single license entitles you to use this script on one website only. A multiple license entitles you to use this script on any number of sites owned or maintained by you.

For support, please visit: <http://formtoemail.com/support/>

If you are new to this, have a look at the "form-to-email primer", here:

http://formtoemail.com/support/form_to_email_primer.php

DESCRIPTION

FormToEmail-Pro is a form processing script written in PHP. It allows you to place a form on your website which your visitors can fill out and send to you. The contents of the form are sent to the email address (or addresses) which you specify below. The form (code supplied below) allows your visitors to enter their name, email address and comments. You can add additional fields to the form or use the script to process any other form, see below.

The script is very secure. Your visitors (and spambots) cannot see your email address. The script cannot be hijacked by spammers for using as a relay to send spam from. It has a number of features to combat contact-form spam.

You can add additional fields to your form, which this script will also process without making any additional changes to the script. You can also use it to process your own existing forms, if doing so, make sure your form action is correct, like so: `<form action="formtoemailpro.php" etc>`. The script will handle the "POST" or "GET" methods. It will also handle multiple select inputs and multiple check box inputs. If using these, you must name the field as an array using square brackets, like so: `<select name="fruit[]" multiple>`. The same goes for check boxes if you are using more than one with the same name, like so: `<input type="checkbox" name="fruit[]" value="apple">Apple<input type="checkbox" name="fruit[]" value="orange">Orange<input type="checkbox" name="fruit[]" value="banana">Banana`

**** PLEASE NOTE **** If you are using the script to process your own forms (or older FormToEmail forms) you must ensure that the email field is named correctly in your form, thus: `<input type="text" name="email">`. Note the lower case "email". If you don't do this, you won't be able to see who the email is from and the script won't be able to check the validity of the email. The autoresponder won't work either. If you are using the form code below, you don't need to check for this.

The script will handle file uploads of any file type and send them as attachments to the email that is sent to you or upload them to your server. There is no limit to the number of files you can upload. To allow your form to upload files, you must put this `enctype="multipart/form-data"` in the form action and use the "post" method, like so: `<form`

action="formtoemailpro.php" method="post" enctype="multipart/form-data">. You must also use a file input <input type="file" name="whatever"> on your form, one for each file you wish the visitor to upload. Give each one a different name. You do not need to make any configuration changes to the script to allow file uploads to be sent as attachments, they will be detected automatically. If you wish to save the uploaded files to your server instead of sending them as attachments then you must enable this in the configuration options below.

This is a PHP script. In order for it to run, you must have PHP (version 4.1.0 or later) on your webhosting account, and have the PHP mail() function enabled and working. If you are not sure about this, please ask your webhost about it.

SETUP INSTRUCTIONS

Step 1: Put the form code on your webpage (if not using an existing form)

Step 2: Enter your email address

Step 3: Configure the script (optional)

Step 4: Upload the files to your webspace

Step 1:

Put the form code on your webpage (if not using an existing form).

If using an existing form, remember to change the action to: <form action="formtoemailpro.php" method="post">. Remember also to use "email" lower case for the email input, like so: <input type="text" name="email">

The HTML code below is for the form that will appear on your form page. This code makes a basic contact form with name, email and comments. Copy the code as it is, and paste it into your webpage:

```
<form action="formtoemailpro.php" method="post">
<table border="0" style="background:#ecec" cellspacing="5">
<tr align="left"><td>Name</td><td><input type="text" size="30" name="name"></td></tr>
<tr align="left"><td>Email</td><td><input type="text" size="30" name="email"></td></tr>
<tr align="left"><td valign="top">Comments</td><td><textarea name="comments" rows="6"
cols="30"></textarea></td></tr>
<tr align="left"><td>&nbsp;</td><td><input type="submit" value="Send"></td></tr>
</table>
</form>
```

Step 2:

Enter your email address.

Enter the email address below to send the contents of the form to. Like this for example:
\$my_email = "webmaster@example.com"; If desired, you can enter more than one email
address separated by commas, like so: \$my_email =
"bob@example.com,sales@example.co.uk,jane@example.com";

*/

```
$my_email = "";
```


/*

The script will now work. Everything below this line is optional. If you wish, you can now proceed to Step 4 (Upload the files to your webspace).

Step 3:

Configure the script (optional).

You can edit these options at any time. If you do so, remember to save the file and upload it to your webspace.

Bcc email addresses. The script can send a Blind Carbon Copy (BCC) of the email to the address(es) specified below. Enter the address(es) between the quotes below, like this for example: `$bcc = "support@example.com,mail@example.com";`

*/

`$bcc = "";`

/*

Subject line. The default subject line for the email that is sent to you, is shown below. To change this, enter your own subject between the quotes. Make sure that you use a subject that will get past any spam filters you might use.

*/

```
$subject = "Google Places and Web Page";
```

```
/*
```

From: email address. Only do this if you know you need to. By default, the email you get from the script will show the visitor's email address as the From: address. In most cases this is desirable. On the majority of setups this won't be a problem but a minority of hosts insist that the From: address must be from a domain on the server. For example, if you have the domain example.com hosted on your server, then the From: email address must be something@example.com (See your host for confirmation). This means that your visitor's email address will not show as the From: address, and if you hit "Reply" to the email from the script, you will not be replying to your visitor. You can get around this by hard-coding a From: address into the script using the configuration option below. Enabling this option means that the visitor's email address goes into a Reply-To: header, which means you can hit "Reply" to respond to the visitor in the conventional way. (You can also use this option if your form does not collect an email address from the visitor, such as a survey, for example, and a From: address is required by your email server.) The default value is: `$from_email = ""`; Enter the desired email address between the quotes, like this example: `$from_email = "contact@example.com"`; In these cases, it is not uncommon for the From: (`$from_email`) address to be the same as the To: (`$my_email`) address, which on the face of it appears somewhat goofy, but that's what some hosts require.

```
*/
```

```
$from_email = " ";
```

```
/*
```

Continue link. You have the option of presenting your visitor with a standard "thank you" message or automatically redirecting them to a "thank you" page (or any page of your choosing) after they have submitted the form. The default is the standard "thank you" message. If you are using the standard "thank you" message, they will be provided with a "continue" link to your homepage (or any page of your choosing) after submitting the form, which they can click on to

continue. Enter the continue link to offer the user. If you do not change this, your visitor will be given a continue link to your homepage. If you are using the automatic redirect option below, the \$continue value is ignored.

If you do change it, remove the "/" symbol below and replace it with the address of the page to link to, eg: "mypage.htm" or "http://www.elsewhere.com/page.htm"

```
*/
```

```
$continue = "http://www. seo.com/merci.htm";
```

```
/*
```

Automatic redirect. If you would like the visitor to be automatically redirected to another page (or site) when they successfully submit the form (as opposed to getting the standard "thank you" message) change the value below to 1, like so: \$auto_redirect = 1;

```
*/
```

```
$auto_redirect = 1;
```

```
/*
```

Enter the web address of the page that you would like the visitor to be automatically redirected to (between the quotes below). You only need to do this if you have enabled the auto redirect above. You can redirect them to any page on any site, it doesn't have to be a page on your own site. It is advisable to write the full URL like so: \$redirect_url = "http://example.com/thankyou.htm";

```
*/
```

```
$redirect_url = "http://www.seo.com/merci.htm";
```

```
/*
```

HTML template for the "thank you" message. Instead of using the standard "thank you" message or redirecting to another page, you can use an HTML template to display a "thank you" message of your own design. This "thank you" page will be displayed on the screen when the form is submitted successfully. The biggest advantage of this method is that you can include submitted values from your form in the page, and you can make the page EXACTLY as you want it. To use this option, you need to make an HTML template file, which you upload to your webspace, then enable the option below. The HTML template file is a normal HTML file, with the addition of some, none, or all of the submitted form values in it. If it suits your purpose, you can make the file without showing any of the submitted form values. Write the file as you would any HTML page. Where you want to show submitted form values in the file, use `ff<form_field>` (replace "form_field" with the name of the submitted form field you want to show). Like this for example:

```
<p><b>State:</b> ff<state></p>
```

The above example will print your visitor's state as entered in the state input `<input type="text" name="state">` on your form (for example).

Save the HTML template file by the name shown below in the `$thank_you_message_template_filename` option, then upload it to your webspace in the same directory/folder as the script (default location for it).

For more information on this, see the section below about using an email template, which works in the same way.

By default, this option is not enabled: `$thank_you_message_template = 0;` To enable it, change the 0 to 1 below, like so: `$thank_you_message_template = 1;`

```
*/
```

```
$thank_you_message_template = 0;
```

```
/*
```

"Thank you" message template filename. This option allows you to state a name and location for the HTML "thank you" message template file. You must name your template file as stated below, and upload it to your webspace in the location stated below. The default is:

`$thank_you_message_template_filename = "thank_you_message_template.php";` This means you need to save your HTML template file as "thank_you_message_template.php" and upload it to the same directory/folder as the script. You can call the file by any name, and you can upload it to any location on your server. If you are keeping the file in the same directory/folder as this script and would like to change the name of the file, you do not need to enter a path to the file, just change the name below, between the quotes, and save your template file by the same name. If you are uploading your template file to a directory/folder other than the one the script is in, then you must state the path, like this for example:

`$thank_you_message_template_filename =
"/home/bob/includes/thank_you_message_template.php";` This configuration is ignored if you have not enabled the `$thank_you_message_template` option above.

```
*/
```

```
$thank_you_message_template_filename = "";
```

```
/*
```

Pre-populate the form. If enabled, a session will be created and the submitted values from the form will be stored in the session. This means the values will be available to your form page should your visitor go back to your form after getting an error. You can then display (pre-populate) the submitted values on the form. To display the values, you need to add some code to your form page (the script can't do that for you) and enable the option below. This prevents the visitor getting a blank form when they return to the form to correct any errors. Such blank forms are caused by browser behaviour and settings (not the script). Some browsers refresh the page when you hit "back" others don't. If the form page is refreshed, the visitor gets a blank form and has to enter their information over again. Enabling the option below and coding your form page to display the submitted values means your visitor won't be faced with a blank form if they have to return to it, they will get the values they submitted.

To display the submitted form values, your form page needs to be a PHP page, with a .php extension, like contact.php for example. If it's an HTML page, just save it with a .php extension.

Add this code to your form page:

Put this at the top:

```
<?php
```

```
session_start();
```

```
if(isset($_SESSION['submitted_form_values'])){extract($_SESSION['submitted_form_values']);}
```

```
?>
```

To display the visitor's entered values on the form (instead of presenting a blank form) you need to put some PHP code into your HTML form code. For each form input on your form (where you expect a visitor to enter data, not a hidden field for example) you need to have a piece of code like this:

```
<?php if(isset($form_field_name)){print stripslashes($form_field_name);}else{print "";} ?>
```

It is used to print a value in the form field. `$form_field_name` is replaced with a variable that is EXACTLY the same name as the form field. Here's an example. Suppose you have an input on your form, like this:

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

To show the visitor's entered value, put the above PHP code into a value for the field, like so:

```
<input type="text" name="Street_Address" value="<?php if(isset($Street_Address)){print stripslashes($Street_Address);}else{print "";} ?>">
```

Note that "`$Street_Address`" is used (two instances) in place of the generic "`$form_field_name`". Do the same for all text inputs, replacing `$form_field_name` with the exact (case sensitive) variable name of the form field (the preceding dollar sign makes it a variable).

The above instructions give you the rudiments of it. For more information about coding for drop-down lists, textareas, radio buttons, checkboxes and a sample basic form page (name, email, comments) correctly coded, please see this page on the support section:

http://formtoemail.com/support/pre_populate_form.php

Next, enable the option below. Default value: `$pre_populate_form = 0`; To enable the option, change the 0 below to 1, like so: `$pre_populate_form = 1`;

```
*/
```

```
$pre_populate_form = 0;
```

```
/*
```

Show errors on form page. By default, the script will issue any error messages on the screen after the form is submitted. The form is not shown, and the user has to hit "back" to get back to the form (the standard error message page can be customized, have a look at this page on the support section for more information:

http://formtoemail.com/support/customizing_error_messages.php).

If you would like the user to be returned to your form page when errors occur and have the error messages shown along with the form, you will need to add some code to your form page to display the errors (the script can't do that for you). You will also need to enable the `$show_errors_on_form_page` option, and enter a value for the `$form_page_url` variable below. Your form page needs to be a PHP page, with a `.php` extension, like `contact.php` for example. If it's an HTML page, just save it with a `.php` extension. By displaying the errors on the form page, it will appear to the visitor that they have not left the page.

Add this code to your form page:

Put this at the top (if "`session_start();`" is not already there):

```
<?php
```

```
session_start();
```

```
?>
```

Then put this code on your page where you would like the errors to appear:


```
<?php

if(isset($_SESSION['formtoemail_form_errors']))

{

print "<div style=\"color:#ff0000\">";

foreach($_SESSION['formtoemail_form_errors'] as $form_error_value){print "<b>" .
stripslashes($form_error_value) . "</b><br>";}

print "</div>";

unset($_SESSION['formtoemail_form_errors']);

}

?>
```

It is recommended that you enable the `$pre_populate_form` option above and add the appropriate code for it in your form page (see above). This means the form values will be retained in the form in the event of an error. Otherwise, a blank form will be shown. You can get code for a basic contact form (name, email, comments) containing the above code to show the errors and the code for displaying the submitted form values. See this page on the support section:

http://formtoemail.com/support/show_errors_on_form_page.php

Next, enable the option below. Default value: `$show_errors_on_form_page = 0;` To enable the option, change the 0 below to 1, like so: `$show_errors_on_form_page = 1;`

```
*/
```

```
$show_errors_on_form_page = 0;
```

```
/*
```

Form page URL. This value is only required if you have enabled the `$show_errors_on_form_page` option above (and have added the appropriate code to your form) otherwise it is ignored. This is the URL of your form page. Your visitors will be returned to it in the case of errors. Default value: `$form_page_url = "";` Enter the full URL below, between the quotes, like this example: `$form_page_url = "http://example.com/contact.php";`

```
*/
```

```
$form_page_url = "";
```

```
/*
```

You can compare two email address inputs to confirm the visitor's email address. The visitor must correctly enter their email address in the two inputs, otherwise they will get an error and the email will not be sent. If using this option, you must have two email inputs on your form named "email" and "email2", like this: `<input type="text" name="email"> <input type="text" name="email2">`. By default, this is not enabled. Default value: `$confirm_email_address = 0;` To enable it, change the 0 below to 1 like so: `$confirm_email_address = 1;`

```
*/
```

```
$confirm_email_address = 0;
```

```
/*
```

Required fields. You can choose to require specific fields on your form. If these are empty when the form is submitted the user will be issued with a message to go back and enter a value. The form will not be sent until the required fields are filled out. If you do not enable required fields, the script will still issue an error if all fields are blank, and will not send an email. To enable checking for required fields, change the value below from 0 to 1, like so: `$required_fields_check = 1;`

```
*/
```

```
$required_fields_check = 0;
```

```
/*
```

Specify which fields to require. You only need to do this if you have enabled the checking for required fields above. Enter the name of the field in quotes. If you wish to require more than one field, separate them with commas. Like this for example: `$required_fields = array('name','comments');` The field name must be exactly as it appears on the form. For example if you have this on your form: `<input type="text" name="Zip_Code">` and you want to require the name and zip code you would do this: `$required_fields = array('name','Zip_Code');` Please note, if you are using an array for an input name (e.g `name="product[]"`) and you wish to make this a required field, enter the name below without the square brackets like so ('product'). The standard form has the three fields shown below. If you have not enabled checking for required fields (above) you can leave the line below as it is, as it will be ignored.

```
*/
```

```
$required_fields = array('name','email','comments');
```

```
/*
```

IP address display. If you enable this, the email sent to you will have the sender's IP address shown at the bottom. If you have enabled the option to write to a CSV file on the server or to attach a CSV file to the email, the Sender's IP address will be shown in these. In all cases, the Sender's IP address is removed from the autoresponse email to the sender. To enable this option, change the 0 to 1, like so: `$show_ip = 1;`

```
*/
```

```
$show_ip = 0;
```

```
/*
```

Check for banned IPs. You can ban specific IP addresses or IP address classes from using your form. To enable checking for these, change the value below to 1, like so: `$banned_ips_check = 1;`

```
*/
```

```
$banned_ips_check = 0;
```

```
/*
```

Banned IPs. Make a list of the IP addresses (or classes) you would like to ban. You must enable checking for banned IPs above, for this to work. Enter the IP address in quotes. For more than

one address, separate them with commas as in this example: `$banned_ips = array('44.22.233.21','82.24.56.1','224.130.87.113');`

To ban an IP address class, enter the class like so (remembering to put a period at the end): `$banned_ips = array('44.','82.24.','224.130.87.');` So looking at the last class in this example, any IP address in the range 224.130.87.0 to 224.130.87.255 will be banned. In other words, any IP address beginning with 224.130.87. will be banned.

Here's an example of individual IP addresses and an IP class (they all go in the same array): `$banned_ips = array('44.22.233.21','224.130.87.113','82.24.');`

Please note, the IP addresses and classes shown in the above examples are fictitious. There is virtually no limit to the number of IP addresses you can list here but the more you have the slower the script will run (as with any script) but you would need to have quite a large list to notice any difference.

The default value is: `$banned_ips = array();` Enter the IP addresses (or classes) between the brackets, if required.

```
*/
```

```
$banned_ips = array();
```

```
/*
```

Banned IP message. If the form is submitted from a banned IP address, the (contents of the form) email will not be sent to you. The script will issue an error to the user. You can edit the error message by changing the message between the quotes below. If you don't want to reveal that the IP address is banned, you could have a message like "An error has occurred. The form was not sent."

```
*/
```

```
$banned_ip_message = "Your IP address is banned. The form was not sent.";
```

```
/*
```

Cookie requirement. If you enable this the script will first check for the existence of a cookie. If the cookie does not exist, the script will exit, an error message will be issued and the email will not be sent. This is a very useful feature as it will block some spam bots. If this is enabled, the user (your website visitor) must have cookies enabled on their browser. To use this feature you must set a cookie on the page that the form is on. If it is on an HTML page you can set the cookie by putting this line in the <head></head> section: <meta http-equiv="Set-Cookie" content="formtoemailpro=1">

If the form is on a PHP page, you can set the cookie by entering this code on the page (before any output - usually near the top): `setcookie('formtoemailpro',1);`

To require a cookie, change the value to 1, like so: `$require_cookie = 1;`

```
*/
```

```
$require_cookie = 0;
```

```
/*
```

Referrer check. By default the script will check for a referrer (referer) header to establish that the form is being submitted from your site. To disable this, change the value to 0, like so: `$check_referrer = 0;`

```
*/
```

```
$check_referrer = 1;
```

```
/*
```

Word block. You can block any word (rude or otherwise), character, string or phrase from being submitted. If such an instance is detected the script will issue an error and the email will not be sent. The script performs this check by default and checks all fields for the words. The default value is: `$word_block = 1;` To disable it, change the value below to 0, like so: `$word_block = 0;`

```
*/
```

```
$word_block = 0;
```

```
/*
```

Blocked words. This is a list of words or characters that will be checked for if word block is enabled above. You can add or remove words from the array below. You can enter a single character or a string of characters or words (it is not case-sensitive). Place each word (or phrase) in quotes and separate them with commas. By default the script will not allow web addresses (URLs beginning with `http://` or `https://`) to be posted. If someone is trying to spam you, you can be fairly certain that they will include a web address in the comments or elsewhere. If you expect your visitors to enter web addresses, then you should remove `'http://','https://'` from the array (or disable word block above). The line below is only used if word block is enabled, otherwise it is ignored. Default value is: `$blocked_words = array('http://','https://','viagra');`

```
*/
```

```
$blocked_words = array('http://','https://','viagra');
```

```
/*
```

Gibberish check. Frequently, contact form spam may consist of gibberish "words" like "YwfZCvLdDQYDTzGAH" or "MldMtrPAgZq" for example. The gibberish check identifies such "words", blocks them and issues an error. If an error is issued, the script will not send the email. It looks for a series of consecutive consonants or consecutive vowels. These "words" very often have a long series of consecutive consonants or consecutive vowels that you would not find in a normal word. You can set a threshold (below) which will allow consecutive vowels or consecutive consonants up to the threshold limit. This check only works for English alphabet vowels and consonants. By default it is enabled. The default value is `$gibberish_check = 1`; To disable it, change the 1 below to a 0, like so: `$gibberish_check = 0`;

```
*/
```

```
$gibberish_check = 1;
```

```
/*
```

Gibberish threshold. The gibberish threshold is used in conjunction with the gibberish check above. It defines a threshold for the number of consecutive consonants or consecutive vowels you will allow in submitted words. The default setting is `$gibberish_threshold = 6`; This means that if any 6 (or more) consecutive vowels or any 6 (or more) consecutive consonants are present in a word, then it will be blocked. Change the value to suit. 6 appears to be a good setting as there are not many English words that have six or more consecutive vowels or consonants. This setting has no effect if the gibberish check is disabled above.

```
*/
```

```
$gibberish_threshold = 6;
```



```
/*
```

Gobbledegook check. You sometimes get spam consisting entirely of gobbledegook characters (Å ð ç etc). These occur when a spammer takes a guess at encoding and gets it wrong. This check will prevent any messages getting through if gobbledegook is found. By default this check is not done. The default value is: \$gobbledegook_check = 0; To enable it, change the value to 1, like so: \$gobbledegook_check = 1;

```
*/
```

```
$gobbledegook_check = 0;
```

```
/*
```

Securimage CAPTCHA. This is a third-party CAPTCHA system, written by Drew Phillips from phpcaptcha.org. It's a good system and it's easy to install. It displays a CAPTCHA image on your form and an input box where your visitor verifies the code in the CAPTCHA image. The script then tests the value of the entered code and gives an error if it is incorrect.

To install it, first you need to put the code for it in your HTML form code (to display the captcha on your form). Here is some sample HTML code for a basic contact form (name, email, comments) including the code for the Securimage CAPTCHA:

```
<form action="formtoemailpro.php" method="post">
<table border="0" style="background:#ecec" cellspacing="5">
<tr align="left"><td>Name</td><td><input type="text" size="30" name="name"></td></tr>
<tr align="left"><td>Email</td><td><input type="text" size="30" name="email"></td></tr>
```

```
<tr align="left"><td valign="top">Comments</td><td><textarea name="comments" rows="6"
cols="30"></textarea></td></tr>
```

```
<tr align="left"><td valign="top">Security code</td><td><br><a href="#"
style="background:#ecec" onclick="document.getElementById('captcha').src =
'/securimage/securimage_show.php?' + Math.random(); return false">Reload
Image</a></td></tr>
```

```
<tr align="left"><td>Verify code</td><td><input type="text" name="captcha_code" size="10"
maxlength="6"></td></tr>
```

```
<tr align="left"><td>&nbsp;</td><td><input type="submit" value="Send"></td></tr>
```

```
</table>
```

```
</form>
```

If you are using an existing form, here is the HTML code you need to add to it:

```
<br><a
href="#" onclick="document.getElementById('captcha').src =
'/securimage/securimage_show.php?' + Math.random(); return false">Reload
Image</a><br><input type="text" name="captcha_code" size="10" maxlength="6">
```

You then need to make a directory/folder in your web space called /securimage and upload the Securimage CAPTCHA files to it (note spelling of "securimage"...only one "e"!). The directory would look like this example:

<http://www.example.com/securimage/>

You then need to download a zip file containing the Securimage CAPTCHA files. Unzip the file and put all the files into your newly created /securimage folder/directory on your web space. You can get the zip file here:

http://formtoemail.com/securimage_captcha_files.zip

(Optionally, you can customize the CAPTCHA image, fonts, colors etc by editing the file /securimage/secureimage.php which is in the files you uploaded to the /securimage directory. For support, see <http://www.phpcaptcha.org/>)

Then all you need to do is enable it below. By default this option is not enabled (\$Securimage_CAPTCHA = 0;). To enable it, change the 0 below to 1, like so:
\$Securimage_CAPTCHA = 1;

```
*/
```

```
$Securimage_CAPTCHA = 0;
```

```
/*
```

reCAPTCHA system. This is a third-party CAPTCHA system (more information here: <http://recaptcha.net/>). reCAPTCHA is a claimed Trademark. If you enable this option, the script will check the user input from the CAPTCHA. To use this system, you first need to register with recaptcha.net and get a public and private key which you enter on your form and in this script (the public key goes in your form page, the private key gets entered in this script below). The keys are only valid for the domain they were registered to (see recaptcha.net for more information).

To install it, first you need to put the code for it in your HTML form code (to display the captcha on your form). Here is some sample HTML code for a basic contact form (name, email, comments) including the code for the reCAPTCHA system. Copy the code below as it is, and paste it into your webpage (you will need to enter your reCAPTCHA public key where indicated, more information below):

```
<form action="formtoemailpro.php" method="post">
```

```
<table border="0" style="background:#ecec" cellspacing="5">
```

```
<tr align="left"><td>Name</td><td><input type="text" size="30" name="name"></td></tr>
<tr align="left"><td>Email</td><td><input type="text" size="30" name="email"></td></tr>
<tr align="left"><td valign="top">Comments</td><td><textarea name="comments" rows="6"
cols="30"></textarea></td></tr>
<tr align="left"><td>&nbsp;</td><td>
```

```
<?php
```

```
require_once('recaptchalib.php');
```

```
$publickey = "enter your reCAPTCHA public key in here between the quotes";
```

```
echo recaptcha_get_html($publickey);
```

```
?>
```

```
</td></tr>
```

```
<tr align="left"><td>&nbsp;</td><td><input type="submit" value="Send"></td></tr>
```

```
</table>
```

```
</form>
```

If using your own form, then you need to paste in this code between the <form></form> tags:

```
<?php
```

```
require_once('recaptchalib.php');
```

```
$publickey = "enter your reCAPTCHA public key in here between the quotes";
```

```
echo recaptcha_get_html($publickey);
```

```
?>
```

Note that you have to enter your reCAPTCHA public key in the above code (same too if using the sample form code above). Your form page MUST be a PHP page for the reCAPTCHA code to work. i.e. your page must have a .php extension like "contact.php" for example. You will also need to download the PHP zip file from recaptcha.net and put the file "recaptchalib.php" in the same directory/folder as the form page and script. The zip file from recaptcha.net has several files in it but "recaptchalib.php" is the only one you need. You can get it here: <http://code.google.com/p/recaptcha/downloads/list?q=label:phplib-Latest>

All the code you need is included in the script. You do not need to get any code from recaptcha.net. The only thing you need from them is your public and private keys and the zip file containing `recaptchalib.php`.

By default this option is not enabled (`$reCAPTCHA = 0;`). To enable it, change the 0 below to 1, like so: `$reCAPTCHA = 1;`

```
*/
```

```
$reCAPTCHA = 0;
```

```
/*
```

Private key for the reCAPTCHA system. Enter your reCAPTCHA private key between the quotes below. You get this from recaptcha.net when you register with them, see above. The key is only valid for the domain it was registered to. This is only used if you have enabled the reCAPTCHA system above. Default is: `$privatekey = "";`

```
*/
```

```
$privatekey = "";
```

/*

identiPIC photo CAPTCHA system. This is a system whereby the visitor has to identify a picture or pictures on your form to proceed. This is to stop spam bots. To use this you need to have the correct code on your form and upload the required images to your web space in the same directory(folder) as the webpage containing your form. If enabled, the script will check the input and issue an error if the options selected do not match the solutions (below). It is advisable to customize this to be unique to you. See <http://identipic.com> for more information.

Here is sample HTML code for a basic contact form (name, email, comments) including the code for the identiPIC photo CAPTCHA. Copy it and paste it into your webpage:

```
<form action="formtoemailpro.php" method="post">
<table border="0" style="background:#ecec" cellspacing="5">
<tr align="left"><td>Name</td><td><input type="text" size="30" name="name"></td></tr>
<tr align="left"><td>Email</td><td><input type="text" size="30" name="email"></td></tr>
<tr align="left"><td valign="top">Comments</td><td><textarea name="comments" rows="6"
cols="30"></textarea></td></tr>
<tr><td>&nbsp;</td><td>
<table border="0">
<tr><td colspan="3">Security test. Please identify the pictures:</td></tr>
<tr><td></td><td></td><td></td></tr>
<tr><td>
<select name="identiPIC_selected[1]">
<option value="">Click to identify</option>
<option>Apple</option>
<option>Cat</option>
<option>Clock</option>
```

```
<option>Dog</option>
<option>Flower</option>
<option>Fork</option>
<option>Hammer</option>
<option>Key</option>
<option>Ship</option>
<option>Tree</option>
</select></td><td>
```

```
<select name="identiPIC_selected[2]">
<option value="">Click to identify</option>
<option>Apple</option>
<option>Cat</option>
<option>Clock</option>
<option>Dog</option>
<option>Flower</option>
<option>Fork</option>
<option>Hammer</option>
<option>Key</option>
<option>Ship</option>
<option>Tree</option>
</select></td><td>
```

```
<select name="identiPIC_selected[3]">
<option value="">Click to identify</option>
```

```
<option>Apple</option>
<option>Cat</option>
<option>Clock</option>
<option>Dog</option>
<option>Flower</option>
<option>Fork</option>
<option>Hammer</option>
<option>Key</option>
<option>Ship</option>
<option>Tree</option>
</select></td></tr></table></td></tr>
```

```
<tr align="left"><td>&nbsp;</td><td><input type="submit" value="Send"></td></tr>
</table>
</form>
```

If you are using your own form you will need to add the HTML code for the identiPIC photo CAPTCHA to it. Use this code and paste it into your existing HTML form code:

```
<table border="0">
<tr><td colspan="3">Security test. Please identify the pictures:</td></tr>
<tr><td></td><td></td><td></td></tr>
<tr><td>
<select name="identiPIC_selected[1]">
<option value="">Click to identify</option>
```



```
<option>Apple</option>
<option>Cat</option>
<option>Clock</option>
<option>Dog</option>
<option>Flower</option>
<option>Fork</option>
<option>Hammer</option>
<option>Key</option>
<option>Ship</option>
<option>Tree</option>
</select></td><td>
```

```
<select name="identiPIC_selected[2]">
<option value="">Click to identify</option>
<option>Apple</option>
<option>Cat</option>
<option>Clock</option>
<option>Dog</option>
<option>Flower</option>
<option>Fork</option>
<option>Hammer</option>
<option>Key</option>
<option>Ship</option>
<option>Tree</option>
</select></td><td>
```

```
<select name="identiPIC_selected[3]">
<option value="">Click to identify</option>
<option>Apple</option>
<option>Cat</option>
<option>Clock</option>
<option>Dog</option>
<option>Flower</option>
<option>Fork</option>
<option>Hammer</option>
<option>Key</option>
<option>Ship</option>
<option>Tree</option>
</select></td></tr></table>
```

You next need to upload the identiPIC photo CAPTCHA images to your webspace. Put them in the same directory/folder as your form page. If using the sample code above, you will need three images: identiPIC_1.jpg, identiPIC_2.jpg and identiPIC_3.jpg . You can get them in a zip file, here:

http://formtoemail.com/identiPIC_images.zip

To enable this option, change the value below from 0 to 1, like so: \$identiPIC_photo_CAPTCHA = 1;

*/

```
$identiPIC_photo_CAPTCHA = 0;
```

```
/*
```

identiPIC photo CAPTCHA solutions. If you change a picture on your form and it has a new solution, enter the correct selection option for the picture below (if using the sample code and images above, leave these settings as they are). The option must be written below exactly as it appears on the select option drop-down list for the appropriate picture on the web form. Use alphanumeric characters (letters and/or numbers) you might get an error otherwise. Spaces are ok but don't have any at the beginning or end of your words. Suppose your first picture is a picture of a bicycle, the lines below would look like this:

```
$identiPIC[1] = "Bicycle";
```

```
$identiPIC[2] = "Flower";
```

```
$identiPIC[3] = "Fork";
```

IMPORTANT. The number of lines must equal the number of identiPIC pictures on your form, otherwise the script will exit. This is a security measure. You can have as many pictures as you wish but there must be an accompanying solution for each one below and an accompanying drop-down list on your form, containing the correct solution. So if (for example) you were to add an identiPIC picture of a ship to your form, making four pictures on your form, the lines would look like this:

```
$identiPIC[1] = "Apple";
```

```
$identiPIC[2] = "Flower";
```

```
$identiPIC[3] = "Fork";
```

```
$identiPIC[4] = "Ship";
```

The lines to edit are the lines below here:

```
*/
```

```
$identiPIC[1] = "Apple";
```

```
$identiPIC[2] = "Flower";
```

```
$identiPIC[3] = "Fork";
```

```
/*
```

Email template. By default the script will send you an email in a standard format. Be it plain text or HTML, you will get pretty much what the script gives you, style-wise. You can make some adjustments/configurations to it, but if you want your email to look a certain way or include custom text, images, messages, styling, layout etc, then you can use an email template. That way, the email you get from the script will be exactly as you want it. To use a template, you need to enable the \$email_template option below and make a template file which you must upload to your server/webpace. You can make your email template plain text or HTML. If using an HTML email template, you must also enable the \$html_format option below. The template is only used for the body of the email, you must set the subject above, as desired. For information about making a template, please see this support page:

http://formtoemail.com/support/email_templates.php

This is the basics. The email template file is a text file written in plain text or using HTML code. The script reads it and uses the information to write the body of the email. You write it just as you wish the email to be written. You use ff<form_field> to get values from your form into the email (replace form_field with the form field name you wish to use). You save the file by the name stated below and upload it to your webpace in the same directory/folder as this script. Here's an example. You have an input on your form called "first_name". You could have this plain text code in your template:

Contact from ff<first_name>.

In the email you receive, `ff<first_name>` will be replaced by the name entered in the "first_name" field on your form by your visitor.

To make an HTML email template, just write the file as if it was an HTML page and use `ff<form_field>` to get values from your form.

To enable the email template option, change the 0 below to 1, like so: `$email_template = 1;`

Remember to enable the `$html_format` option below if you are using an HTML template.

```
*/
```

```
$email_template = 0;
```

```
/*
```

Email template filename. This option tells the script the name of the email template file to use (if enabled above). The default is this: `$email_template_filename = "email_template.php";` but you can call the file by any name. Enter the name below for your email template file if calling it by a different name. Enter it between the quotes. Otherwise, just keep the name as it is and use it for your template file. If your template file is in the same directory/folder as this script, then you only need to enter the filename. If your template file is in another directory/folder then write the full path and filename, like this example: `$email_template_filename = "/home/bob/includes/email_template.php";`

```
*/
```

```
$email_template_filename = "email_template.php";
```

```
/*
```

HTML formatting. By default (if not using an email template) the script will send you an email in plain text but you can choose to have the output sent to you in HTML format. This presents the output in a table with different colors and bold fonts which makes it very neat and easy to read. You can customize it below. Enabling this option also means that the autoresponder email (if enabled) will also be in HTML format. If using an HTML email template, then you should also enable this option. To enable HTML formatting change the value below to 1, like so:

```
$html_format = 1;
```

```
*/
```

```
$html_format = 0;
```

```
/*
```

HTML output customizing. Change the values below, as required, to customize the HTML output. These values are only used if HTML formatting is enabled above, otherwise they are ignored. These values are only applied to the HTML email produced by the script. If using your own HTML email template (see above) these values are ignored. You can use words (red, green) hexadecimal or RGB values for the colors. The table outputs two columns. The left one contains the keys (variables from your form fields) and the right one contains the values entered by your visitor. Try it with the default settings first so you can see what it looks like. The HTML format uses "old-fashioned" HTML (bgcolor,) because some email readers have difficulty with CSS.

You can also change the font settings for the autoresponder message below (if enabled).

```
*/
```

```
$table_cellpadding = "5";
```

```
$table_cellspacing = "1";  
$table_background_color = "#000000";  
$table_left_column_color = "#ecec";  
$table_left_column_font = "arial";  
$table_left_column_font_size = "2";  
$table_left_column_font_color = "#000000";  
$table_right_column_color = "#ffffff";  
$table_right_column_font = "arial";  
$table_right_column_font_size = "2";  
$table_right_column_font_color = "#000000";
```

```
$autoresponder_font = "arial";  
$autoresponder_font_size = "2";  
$autoresponder_font_color = "#000000";
```

```
/*
```

Character set. Specify a character set (encoding) to be used in the output email and the built-in "thank you" message. Only change this if you need to and you know the correct one to use. Enter the desired value between the quotes. Default: `$character_set = "iso-8859-1"`; Please also state the character set in the <head> of your form page. So for example if you are using utf-8, the line in your form page would be: <meta http-equiv="Content-Type" content="text/html; charset=utf-8"> and the variable below would be: `$character_set = "utf-8"`;

```
*/
```

```
$character_set = "utf-8";
```

```
/*
```

Encode From: name and subject line. Apply encoding to the email From: name and the email subject line. Only do this if you need to. You will normally only need to do this if you are using exotic characters that are getting garbled in the From: name or subject line. It uses the \$character_set variable value above. The default setting is disabled, thus: \$encode_name_subject = 0; To enable it, change the 0 to 1, like so: \$encode_name_subject = 1;

```
*/
```

```
$encode_name_subject = 0;
```

```
/*
```

CSV attachment. You can choose to have the form data included in a CSV (Comma Separated Value) file that is sent as an attachment to the email the script sends (you still get the form data in the email). The file contains the field names as the header row and the form data as the second row. The file only contains data from the submitted form and does not accumulate data from all form submissions (see below if you want a CSV file containing all form submissions). The CSV file can be opened in Excel, for example. You do not need to make any changes to your form. To enable this, change the value below to 1, like so: \$csv_attachment = 1;

```
*/
```

```
$csv_attachment = 0;
```

```
/*
```


CSV file on server. If you enable this, all data from submitted forms will be accumulated in a CSV file on your server. You can then download the file and open it in Excel (for example) and search/sort/manipulate/print the data to suit. By default the CSV file is kept in the same directory/folder as the script and is named "form_data.csv". You can change the location of the file below. The first line of the file is a header row made up of the field names from your form. The subsequent rows show the data submitted from your form, one line of data for each form submitted.

*** IMPORTANT ***

For this to work, PHP must have permission to write to the directory where the CSV file will be created and kept (the script creates the file automatically if it does not already exist). For Unix/Linux type operating systems, set the permissions for the directory to CHMOD 770. This allows you (the owner) and PHP to write to the directory but does not allow the rest of the world to do so. You do not need to change permissions for the CSV file itself.

To enable this option, change the value below to 1, like so: `$csv_file_on_server = 1;`

```
*/
```

```
$csv_file_on_server = 0;
```

```
/*
```

Path to CSV file on server. This is the path to the location on your server where the CSV file will be kept. By default it is in the same directory/folder as the script. The default setting is: `$path_to_csv_file = dirname(__FILE__) . "/";` You can change the location by entering the full server path below. Like this for example: `$path_to_csv_file = "/home/users/bob/files/";` Or this example for a Windows server: `$path_to_csv_file = "C:/inetpub/wwwroot/csv/";` (Note the use of forward slashes which is acceptable for Windows servers. If you use backslashes you will get errors in the PHP code). Note that you must have a trailing slash at the end. Note also that you do not include the file name in the path.

```
*/
```

```
$path_to_csv_file = dirname(__FILE__) . "/";
```

```
/*
```

CSV filename. Enter a filename below for the CSV file or keep the default name as it is. This name is used for the file that is sent as an attachment and/or the file that is kept on your server. If you enter a name, put it in quotes like this example: `$csv_filename = "sales_enquiry.csv";`
Default value: `$csv_filename = "form_data.csv";`

```
*/
```

```
$csv_filename = "form_data.csv";
```

```
/*
```

Show date and time submitted. If you enable this option, the date and time that the form was submitted, will be shown in the email you receive, in the autoresponse email (if enabled) and in the CSV file output (if enabled). The date and time are taken from your server, where the script and PHP reside. By default it is not enabled (`$show_date_and_time = 0;`). To enable it, change the 0 to 1, like so: `$show_date_and_time = 1;`

```
*/
```

```
$show_date_and_time = 1;
```

```
/*
```

Create a "unique" Message ID. If you enable this option, a "unique" Message ID number will be generated and shown in the body of the output email from the script. You can use this to identify the form submission. Technically this is not a "unique" number, it is a psuedo-random eight-digit number. To make it unique would require the use of a database or a separate file to store previously issued numbers. This is not supported in the script. In practice, the ID will probably be unique and if used in conjunction with other form data, such as the visitor's email address, it should provide a pretty good reference point. By default it not enabled: `$message_id = 0;` To enable it, change the 0 to 1, like so: `$message_id = 1;`

```
*/
```

```
$message_id = 0;
```

```
/*
```

Autoresponder. You can configure the script to send an automated email response to the enquirer. By default, the message will contain a subject line, a header message, the data from the form submitted and a footer message (signature) but you can write your own autoresponse email using a template, see further below. You can also attach a file to the autoresponse, see below. Configure the various options in the lines below.

If HTML output is enabled above, then the autoresponse email will also be in HTML format.

By default the script will not issue an autoresponse. To enable the autoresponder, change the 0 below to 1, like so: `$autoresponder = 1;`

```
*/
```

```
$autoresponder = 0;
```

```
/*
```

By default the From: address shown in the autoresponse email will be the email address that is used to send the contents of the form to, and is configured in the \$my_email variable above.

You can enter a different From: email address below. Like this for example:

```
$autoresponder_from = "support@example.com";
```

```
*/
```

```
$autoresponder_from = "";
```

```
/*
```

Enter a subject line for the autoresponse email below. Keep it between the quotes.

```
*/
```

```
$autoresponder_subject = "Your enquiry";
```

```
/*
```

Enter a message below, between the quotes, that will be shown at the top of the autoresponse email. This is ignored if \$autoresponse_email_template is enabled below.

```
*/
```

```
$autoresponder_header_message = "Thank you for your enquiry. We will get back to you shortly. The data you submitted is shown below.";
```

```
/*
```

Enter a message (or signature) below, between the quotes, that will be shown at the bottom of the autoresponse email. This is ignored if \$autoresponse_email_template is enabled below.

```
*/
```

```
$autoresponder_footer_message = "";
```

```
/*
```

Send an attachment with the autoresponder. You can attach a file to the autoresponse message. To do so, you need to enable it then enter the information about the file, its name, its location and the filetype in the configuration options below. To enable a file to be attached to the autoresponse email, change the 0 to 1 below, like this: \$autoresponder_attachment = 1;

```
*/
```

```
$autoresponder_attachment = 0;
```

```
/*
```

Autoresponder attachment file name. Between the quotes below, enter the name of the file you are attaching, like this for example: \$autoresponder_attachment_file = "brochure.doc";

*/

```
$autoresponder_attachment_file = "";
```

/*

Autoresponder attachment location. In the quotes below, enter the location (path) of the file you would like to attach. It must be the full server path to the file and must have a trailing slash, like this for example: \$autoresponder_attachment_path = "/home/username/files/"; Do not enter the file name here, that is done above.

*/

```
$autoresponder_attachment_path = "";
```

/*

Autoresponder attachment content (MIME) type. Between the quotes below, enter the file content type. PHP cannot tell what the content type is, so you must enter it yourself. These examples are for common content types (Word document .doc, Excel spreadsheet .xls, Adobe PDF document .pdf, JPEG image .jpg .jpeg):

```
$autoresponder_attachment_content_type = "application/msword";
```

```
$autoresponder_attachment_content_type = "application/ms-excel";
```

```
$autoresponder_attachment_content_type = "application/pdf";
```

```
$autoresponder_attachment_content_type = "image/jpeg";
```

You can get a list of MIME content types here:

<http://www.utoronto.ca/webdocs/HTMLdocs/Book/Book-3ed/appb/mimetype.html>

```
*/
```

```
$autoresponder_attachment_content_type = "";
```

```
/*
```

Autoresponse email template. Enabling this option allows you to use a template for the autoresponse email that is sent to the form submitter. This works in the same way as a template for the standard output email from the script (see `$email_template` above) but uses a different template file (although you can use the same file for both if you wish). If using an HTML template (as opposed to plain text), `$html_format` must be enabled above. The template is only used for the body of the email, you must set the subject above, as desired. By default this option is disabled with a value of 0. To enable it, change the value below to 1, like so:

```
$autoresponse_email_template = 1;
```

```
*/
```

```
$autoresponse_email_template = 0;
```

```
/*
```

Autoresponse email template filename. The option below, states the name of the template file to use for the autoresponse email if so enabled. By default it is:

```
$autoresponse_email_template_filename = "email_template_autoresponse.php"; but you can call the file by any name. If using a different filename, put it between the quotes below. If your template file is in the same directory/folder as this script, then you only need to enter the filename. If your template file is in another directory/folder then write the full path and filename, like this example: $autoresponse_email_template_filename = "/home/bob/includes/email_template_autoresponse.php";
```

```
*/
```

```
$autoresponse_email_template_filename = "email_template_autoresponse.php";
```

```
/*
```

Ignoring fields. You might have fields/inputs on your form that you want to ignore, i.e you do not want them sent in the email you receive from the script. Typically this might be a "submit" input. If you wish to ignore certain fields, you must enable this option and list the fields to ignore in the next option below. If using an email template, you don't need to enable this. By default it is enabled. To disable it, change the 1 below to 0, like so: \$ignore_fields = 0;

```
*/
```

```
$ignore_fields = 1;
```

```
/*
```

In the array below, list the field names that you want to be ignored. Enter the names EXACTLY as they appear on your form. An example is shown below. You can edit it as required but remember to put the values in quotes and separate them with commas. The array below is ignored if you have not enabled \$ignore_fields above.


```
*/
```

```
$fields_to_ignore = array('Submit','submit');
```

```
/*
```

Sort fields. You can specify the order that your form fields are displayed in the email sent from the script. To use this option you must enable it then enter the form fields order in the next section below. If using an email template, you can ignore this. To enable it, change the 0 below to 1, like so: `$sort_fields = 1;`

```
*/
```

```
$sort_fields = 0;
```

```
/*
```

Form fields order. In the array below (`$field_order_keys`) you can set the order of your form fields so that they will show in the email in the order that they appear in the array. You must enable `$sort_fields` above for this to take effect, otherwise it is ignored. The array shown below is an example. Enter your form field names in the array. Put them in quotes and separate them with commas as in the example below. You must name the fields EXACTLY as they are named in your form. ONLY the fields in the array will be shown in the email sent from the script and in the order entered in the array. If you have an array input field on your form, like `fruit[]`, then enter the name without the square brackets, like so: `'fruit'`.

```
*/
```

```
$field_order_keys = array('email','comments','name');
```

```
/*
```

Line spacing. If you receive your email in plain text, the form data will be shown on separate lines with a space in between them. This is to make it easy to read. This is default. You can change the output so that the data is shown on new lines but without spaces. To do so, change the value below to 0, like so: `$line_spacing = 0;` This option is ignored if you are using an email template.

```
*/
```

```
$line_spacing = 1;
```

```
/*
```

Show blank fields. By default the script will only return form fields where a value has been entered. If you would like the script to return all form fields including blanks, then enable `$show_blank_fields` below. To enable it, change the 0 to 1 like so: `$show_blank_fields = 1;` This option is ignored if you are using an email template.

```
*/
```

```
$show_blank_fields = 0;
```

```
/*
```

Block file types. If you are uploading files from your form, you can block certain file types. To do so, enable this feature then enter the blocked file types in the next option. To enable this

feature, change the 0 below to 1, like so: `$block_file_types = 1;` Alternatively you can specifically state which file types you will allow, see below. Blocking file types will allow ALL file types to be uploaded EXCEPT those that you choose to block.

```
*/
```

```
$block_file_types = 1;
```

```
/*
```

File types to block. List in the array below, the file extensions that you want to block. If the script detects a file with one of these extensions, it will issue an error and stop processing the form. Enter file types in lower case, in quotes and separated by commas, like this default setting: `$file_types_to_block = array('.com', '.bat', '.exe');` Note that you must include the dot (period). You can add or remove file extensions to suit. This array is ignored if you have not enabled `$block_file_types` above so it is safe to leave it as it is. Enter the file type extensions in lower case, ".exe" will block "file.EXE" and "file.exe" for example.

```
*/
```

```
$file_types_to_block = array('.com', '.bat', '.exe');
```

```
/*
```

Allow file types. If you are uploading files from your form, as an alternative to blocking certain file types (above) you can choose to only ALLOW certain file types. To do so, enable this feature then enter the allowed file types in the next option. To enable this feature, change the 0 below to 1, like so: `$allow_file_types = 1;` Stating file types to allow will only allow those that you state below to be uploaded, ALL others will not be allowed. By default, ALL file types are allowed.

```
*/
```

```
$allow_file_types = 0;
```

```
/*
```

File types to allow. List in the array below, the file extensions that you want to allow. If the script detects a file type that is not allowed, it will issue an error and stop processing the form. Enter file types in lower case, in quotes and separated by commas, like this default setting: `$file_types_to_allow = array('.doc','.pdf','.jpg');` Note that you must include the dot (period). You can add or remove file extensions to suit. This array is ignored if you have not enabled `$allow_file_types` above so it is safe to leave it as it is.

```
*/
```

```
$file_types_to_allow = array('.doc','.pdf','.jpg');
```

```
/*
```

Set a maximum file upload size. You can check the size of uploaded files and reject those that exceed a maximum file size set below. This feature is Independent of `$block_file_types` above. It only requires a value to be set for it to be enabled, otherwise the check is not carried out. By default the value is empty and the check is not carried out, this is the default setting: `$max_file_size = ""`; To enable this feature, enter a value between the quotes. The value is the maximum permitted file size in bytes. 1 Megabyte (MB) = 1048576 Bytes. Do not enter punctuation in the value. Here is an example: `$max_file_size = "1048576"`; This check is independent of the check that PHP makes on your server for the size of uploaded files (in `php.ini`). Please note. If an uploaded file exceeds the limit set in `php.ini` then PHP will reject the file and it will not get passed to the script. This means that the script will not issue an error because it knows nothing about the file. If you want the script to issue an error in this case, increase the limit in `php.ini` to a value that is greater than the value you set here.

```
*/
```

```
$max_file_size = "";
```

```
/*
```

Upload files to the server. By default, any files uploaded from your form will be sent as attachments to the email the script sends you. Alternatively you can upload these files to your server. Uploaded files will overwrite any existing files of the same name in the same location. See the section in the description at the top of this script for information about handling file uploads. To enable uploads to your server, change the 0 to 1 in the line below, like so:

```
$upload_files_to_server = 1;
```

```
*/
```

```
$upload_files_to_server = 0;
```

```
/*
```

Upload files directory. By default the uploaded files will go in the same directory/folder that the script is in. You can instead name a directory on your server where you would like the files to go. PHP must have permission to write to the directory. Permissions are down to your server setup but CHMOD 777 will work in every case. If the directory is above the web root, you will get the file-names-only shown in the email from the script. If the directory is web-accessable you will get a link to the uploaded file(s) in the email from the script which you can click on to download them. If using the latter option it is better to enable the HTML output option in the script as it makes displaying the links more predictable (it handles spaces in the file names better). If you specify a directory below, put it in quotes and include a trailing slash (you must use the full server path), as in this example: `$path_to_uploaded_file = "/home/username/uploads/";` Or this example for a Windows server: `$path_to_uploaded_file = "C:/inetpub/wwwroot/uploads/";` (Note the use of forward slashes which is acceptable for Windows servers. If you use backslashes you will get errors in the PHP code).

```
*/
```

```
$path_to_uploaded_file = dirname(__FILE__) . "/";
```

```
/*
```

Prefix for uploaded file names. You can specify a prefix for the uploaded files, this is optional. This will only be applied to files uploaded to your server, not to files sent as attachments. You can hard-code a prefix or you can use a value from your form. To hard-code a prefix, put it in quotes like so: `$uploaded_file_prefix = "user_file_";` This prefix name will be applied to EVERY file uploaded. You might find it useful to prefix the file name with the user's name. You can do so like this: `$uploaded_file_prefix = $_REQUEST['name'];` This assumes that the name input on your form is "name", otherwise change it to suit. You can prefix it with any value from your form, just change 'name' in this example with the name of the field from your form. To make the file name more readable, you can put an underscore after the prefix like so: `$uploaded_file_prefix = $_REQUEST['name'] . "_";` You can use more than one form value and separate them with underscores like so: `$uploaded_file_prefix = $_REQUEST['firstname'] . "_" . $_REQUEST['lastname'];` (The example assumes you have the inputs "firstname" and "lastname" on your form).

```
*/
```

```
$uploaded_file_prefix = "";
```

```
/*
```

Replace spaces in filenames of uploaded files. If files are uploaded to the server, a link is given to them in the output email from the script, allowing the recipient to click on the link to download the file. If the filename contains spaces, the link will be broken when viewing it in plain text. This can be overcome by replacing the spaces with hyphens. Enabling the option below will replace spaces with hyphens in uploaded filenames. Note, this only applies to files

uploaded to the server, not files sent as attachments. By default this option is disabled:
\$replace_spaces_in_filenames = 0; To enable it, change the 0 to 1, like so:
\$replace_spaces_in_filenames = 1;

```
*/
```

```
$replace_spaces_in_filenames = 0;
```

```
/*
```

Show attachments in the email body. If files are uploaded from your form and sent as an attachment to the email from the script, you can choose to show the file names in the body (message) of the email. This is independent of the way the file names are displayed when uploading them to your server (information on that above). This setting only applies to files uploaded and sent as attachments. If enabled it will show the name of the file input from your form and the corresponding file name that was uploaded. Same goes for one or multiple file inputs. By default it is enabled. Default setting: \$show_attachments_in_email_body = 1; To disable it, change the 1 to 0 below, like so: \$show_attachments_in_email_body = 0;

```
*/
```

```
$show_attachments_in_email_body = 1;
```

```
/*
```

End of configuration options.

Step 4:

Upload the files to your webspace.

Save this file (formtoemailpro.php) and upload it together with your webpage (containing the form) to your webspace. Save the file as it is, you don't need to delete the instructions! If you are using your own form, remember to change the form action to formtoemailpro.php - <form action="formtoemailpro.php" etc> and remember to name your email input "email", lower case, like so: <input type="text" name="email">. Put the files in the same directory (folder). You do not need to set permissions for the script.

THAT'S IT, FINISHED!

You do not need to make any changes below this line.

J'avoue être à cours de ressources...