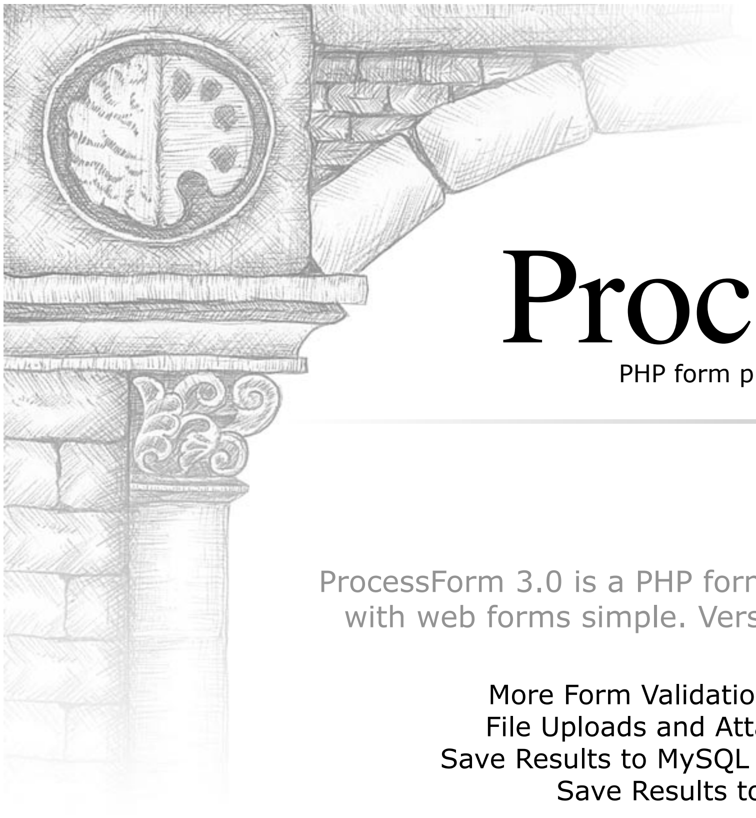


www.mindpalette.com

Web Development Tools | GoLive Actions

ProcessForm 3.0

PHP form processing script by Nate Baldwin, www.mindpalette.com

ProcessForm 3.0 is a PHP form processing script designed to make working with web forms simple. Version 3.0 brings many great features such as...

More Form Validation Options
File Uploads and Attachments
Save Results to MySQL Database
Save Results to Text File

More Configuration Options
More Redirect Page Options
Easier Confirmation Page Styling
...and More!

Contents

CONTINUE >

Getting Started

Your Server

Before spending much time setting this script up, I’d recommend making sure that your server not only supports PHP, but has PHP configured properly to be able to send out emails. In the “Extras” folder included in this download, you will find a “Test PHP Email” utility that you can use to make sure your servers have what they need to use this script.

If your server fails the test, it may just be that your web host requires a different extension for PHP files (such as .phtml or .php3), or they may just need to activate PHP for your account. If things do not seem to be working correctly with the server test, contact your web host and find out if there are any special requirements for using PHP on the server, or for letting PHP send out email messages.

Hidden Form Fields

These instructions are for setting up ProcessForm 3.0 with a standard HTML or text editor. If you are using GoLive or Dreamweaver, check the download for a PDF manual made specifically for your environment.

Many of the features of ProcessForm use “hidden” form fields. This will allow your form to pass the instructions to the script without having the visitor see these fields on your form. If you are not familiar with HTML hidden form fields, the illustrations below will show you the correct syntax.

Hidden Form Field Code:

```
<input type=”hidden” name=”hiddenName” value=”hiddenValue”>
```

The next page will tell you just the basics in setting up the script, and should have you up and running in no time. The following pages will explain all the optional features to fine-tune the script for your site.

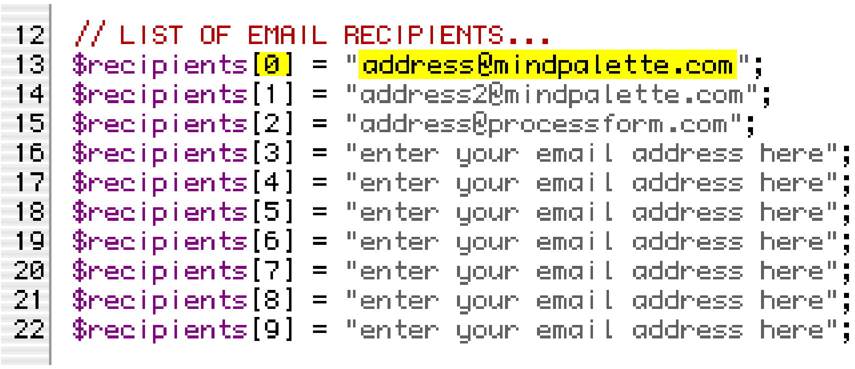
CONTINUE >

Basic Set-up Instructions

One of the problems with many other form processing scripts is that they are vulnerable to spam attacks that hi-jack your script and use it to send out spam email, making it look like YOU sent them. The way we get around that problem is to have a list of recipient email addresses inside the script that ProcessForm is allowed to send messages to.

This list is hidden from email address harvesting scripts, and even if hi-jacked, ProcessForm could only send messages to addressed in this pre-defined list, so it would make it pretty worthless to a spammer. That being said, our first task will be to build the list of “recipient” addresses that ProcessForm is allowed to send mail to.

Open the ProcessForm.php file into your simple text or HTML editor. You’ll find the recipients list starting on line 13. Inside the double quote marks in each line, replace the existing text with an email address. If you don’t need that many addresses, just leave the rest as they are. If you need more, add as many extra lines as you need by copying and pasting, but make sure each $recipients line has a unique key number inside the [brackets].



```
12  // LIST OF EMAIL RECIPIENTS...
13  $recipients[0] = "address@mindpalette.com";
14  $recipients[1] = "address2@mindpalette.com";
15  $recipients[2] = "address@processform.com";
16  $recipients[3] = "enter your email address here";
17  $recipients[4] = "enter your email address here";
18  $recipients[5] = "enter your email address here";
19  $recipients[6] = "enter your email address here";
20  $recipients[7] = "enter your email address here";
21  $recipients[8] = "enter your email address here";
22  $recipients[9] = "enter your email address here";
```

Now you can save and close the ProcessForm.php file, and we’re ready to set up your HTML to work with your script. This tutorial assumes you already have a form page created, or at least know how to build a HTML form.

To use the script, the form will have to know where to submit the form contents to. Set the form action to be the path on your server to get to the ProcessForm.php file. Also, make sure the form method is set to post. When you’re finished, if your script is uploaded to the same directory as your HTML form, the form tag should look something like...

```
<form action=”ProcessForm.php” method=”post”>
```

Now we just have to tell the form which email address to send the results to. To do this, add a hidden field to your form and set the Name to “recipient”, and set the value to be the index number of the address you want the form to send to. The index number is what was inside the [brackets] in the $recipients code we edited earlier in the script file. If you want one form to send to more than one email address, just add as many index numbers as you need, separated by commas.

```
<input type=”hidden” name=”recipient” value=”0,1,4”>
```

CONTINUE >

Test Basic Installation

At this point, the script should work with all its default settings. I'd recommend giving it a test run before spending much more time fine-tuning the optional settings. You will probably need to upload the form and the .php script file to your server in order to test, since PHP scripts are a "server-side" technology and need to be run on a web server.

Fill in your test form and submit it. If the script handles the form correctly (you should see the default "Thank You" confirmation screen) and you receive the resulting email at the address you entered for the recipient, then we're ready to continue with the optional settings. If not, we have a bit of trouble-shooting to do. For help, see the FAQ.pdf file for support information.

Fine-Tune your Settings

There are 2 ways to modify the script and use the built-in features. For all the primary features, you will use hidden form fields to tell the script what to do. There are quite a few hidden form field options, and these settings can be different for every form that uses the same script, so one ProcessForm.php script may be used with several different forms.

Aside from hidden form fields, other settings can be changed inside the script itself. These settings will affect all forms that are submitted to the script. If you decide to change any of these settings inside the script, remember that you are changing programming code, and the "syntax" is very important, so forgetting a quotation mark or semi-colon could easily cause the entire script to stop working. I'd highly recommend making a back-up of the ProcessForm.php file before making any internal changes.

Most of the rest of this tutorial will cover how to use both of these methods to get the script to work best for your needs and to match the look of your site.

CONTINUE >

Basic Hidden Field Settings

subject

```
<input type=”hidden” name=”subject” value=”Web Form Submission”>
```

If you’d like to change the subject of the email that the script sends to your recipient email, then create a new hidden field named “subject” and set the value to whatever you’d like the new email subject to be.

required

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

To make any of your form fields required (the visitor can’t submit the form until the required fields have a value entered), then add a hidden field named “required” and for the value, list the names of the fields that must be filled in before submitting. To require more than one field, add as many field names as you need into the value, separated by commas.

email_only

```
<input type=”hidden” name=”email_only” value=”email”>
```

To require than any of your form fields be a valid email format, add a new hidden field named “email_only” and set the value to the list of form fields that must be in email format, separated by commas. This function will only determine if the address entered is formatted as an acceptable email address, and will NOT determine if the email address actually exists.

numbers_only

```
<input type=”hidden” name=”numbers_only” value=”item_number”>
```

To require that a form field’s value has only numbers entered (for quantities, item numbers, etc.), make a new hidden form field named “numbers_only” and set the value to be the list of form fields whose values can only be numbers, separated by commas. Decimal points or periods are also allowed, and numbers submitted with a comma, such as 2,500, will be considered valid, but the commas will be stripped automatically by the script.

letters_only

```
<input type=”hidden” name=”letters_only” value=”item_name”>
```

To require that a form fields value only be allowed to contain normal upper or lower case letters (a-z, A-Z), then add a hidden field named “numbers_only” and set the value to be the list of fields that can only be letters, separated by commas.

CONTINUE >

lowercase

```
<input type="hidden" name="lowercase" value="email">
```

To have the script convert any field values to all lower case (such as an email field), add a new hidden field named “lowercase” and set the value to be the list of field names to convert to lower case, separated by commas.

uppercase

```
<input type="hidden" name="uppercase" value="item_number, name">
```

This one works just like “lowercase” except the field is named “uppercase” and coverts the field values to all caps.

force_match

```
<input type="hidden" name="force_match" value="password1, password2; email1,
email2, email3">
```

To require that 2 fields have the same value entered before allowing the form to be processed (such as a Password and Confirm Password field), add a new hidden field named “force_match” and set the value to be the fields that must have the same value, separated by commas. If you have more than one set of fields that need to match (like 2 email fields that much match, and 2 password fields that must match), separate the groups of fields by semicolons.

sender_email

```
<input type="hidden" name="sender_email" value="email">
```

To have the “reply to” and “from” address in the email be an email address entered by the person filling out the form, add a new field named “sender_email”, and for the value, enter the name of the other form field that will have the email address of the person completing the form (usually a text field).

sender_name

```
<input type="hidden" name="sender_name" value="your_name">
```

Similar to the “sender_email” function, this option will have the sender name show up as the “from” and “reply to” in the email message. This function can only be used with the “sender_email” function, and will cause the “from” address to look something like... Firstname Lastname <emailaddress> ...in your email application. Set the name of the field to “sender_name” and the value to the name of the other form field that has the name of the sender.

link_url

```
<input type="hidden" name="link_url" value="http://www.mindpalette.com">
```

After the form has been processed, this option will let you add a link to the form results page to get back to your home page (or any other page) after the form has been submitted. Name the field “link_url” and set the value to be the URL of the bage your link returns to. If your form was in a popup window, you can enter “close” for the value to have the script generate a JavaScript “close window” link. Enter “back” for the value if you want the script to generate a JavaScript link to go to the previous page.

link_text

```
<input type="hidden" name="link_text" value="MindPalette Home">
```

This option will determine the text for the link with the “link_url” option above. If not used, the script will default to “back to home”. If the value of link_url is “close”, this will default to “close window”, and if the link_url value is “back”, it will default to “back to form”

CONTINUE >

Basic Hidden Field Settings, continued

exclude

```
<input type=”hidden” name=”exclude” value=”submitButtonName, submitx”>
```

This option will allow you to exclude certain form fields from the resulting email and the confirmation or redirect page (such as excluding your submit button data and any unnecessary information that the form is trying to pass). Add a new hidden field named “exclude” and set the value to be the list of form field names to exclude, separated by commas.

exclude_display

```
<input type=”hidden” name=”exclude_display” value=”password, confirm_password”>
```

Similar to the “exclude” field, this option will exclude any field values from the confirmation or redirect page, but will still send the information in the email message. Add a new hidden field named “exclude_display” and for the value, enter the list of form field names to exclude from the confirmation or redirect page, separated by commas.

exclude_email

```
<input type=”hidden” name=”exclude_email” value=”password, confirm_password”>
```

Similar to the “exclude” field, this option will exclude any field values from email message, but will still send the information to the confirmation or redirect page. Add a new hidden field named “exclude_email” and for the value, enter the list of form field names to exclude from the email message, separated by commas.

recipient_name

```
<input type=”hidden” name=”recipient_name” value=”MindPalette”>
```

In the default confirmation page, there is a message telling the visitor what email address the form results were sent to. If you’d like to not show the email address to avoid getting unwanted messages, and would rather have the script say the name of the person or company that the email was sent to instead of the email address, then add a new form field named “recipient_name” and for the value, enter the name you’d like to have it use.

sort

```
<input type=”hidden” name=”sort” value=”name, phone, email, comments”>
```

This option will let you control the sort order of the form results in the resulting email and the confirmation or redirect page. Add a new hidden field named “sort” and set the value to be the names of your form orders in the order you’d like them to appear. Only the fields found in the sort value will be sent in the email and shown in the confirmation or redirect page.

CONTINUE >

Custom Redirect Pages

If you'd like to have the script redirect to one of your other pages after the form has been successfully processed instead of the default Confirmation / Thank You page, you'll want to use one of the built-in redirect options.

redirect

```
(moved to script configuration settings, line 46)
```

the script to use your redirect page. Name the field "redirect" and for the value, enter the URL address of the page you'd like to redirect to. For best result, use a relative URL instead of an absolute URL. It's also best to put the redirect page into the same folder as the ProcessForm.php script page. If that's the case, you can just enter the name of the page, as shown in the example above.

If you don't need the form results to display in your redirect page, then we're done. Usually, though, you'll want to tell the visitor what information was sent with the form as a confirmation. The challenge is getting the form information passed to your own page so it can be displayed. There are 2 methods included in the script.

redirect_type

```
<input type="hidden" name="redirect_type" value="include">
```

To tell the script which type of redirect to use, add another hidden field named "redirect_type", and the value will either be set to "include" or "query". If no value is entered, or if the "redirect_type" field is not used, the script will default to the "include" method. Both methods are explained in detail below.

"include" redirects

The preferred redirect method is "include" (the default). This method actually imports your redirect page into the script page, so it's important that your put your redirect page in the same folder as the script. It's also important to make sure your redirect page has a .php extension (or whatever other extension your server requires to recognize a file as a PHP page). Build your page however you'd like it to look, then switch to source view in GoLive, and go to the area you'd like the form results to print to. Sometimes it's easier to make a page break <br> tag in layout mode, then select it and switch to source mode, if you have trouble finding your place in source. Then, insert this PHP code...

<?php @redirectHTML("fieldNameCSS", "fieldValueCSS"); ?>

Make sure you type it in exactly as shown. Then, if you'd like to use CSS classes to control the way the form field names and values look, make your CSS classes, and replace the "fieldNameCSS" and "fieldValueCSS" with the CSS class name you'd like it to use. Make sure to use the quotation marks as shown. Alternately, you could just name your CSS classes .fieldNameCSS and .fieldValueCSS and not have to change the PHP code.

CONTINUE >

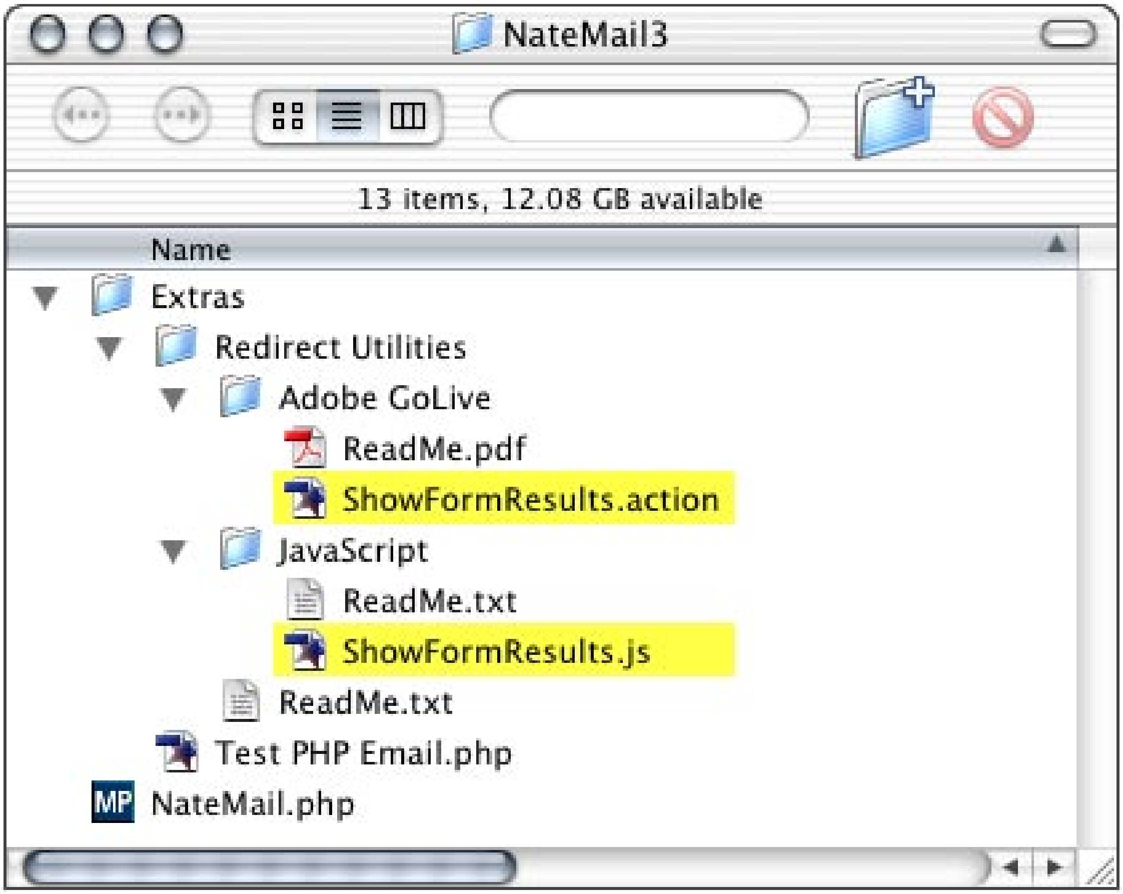
Custom Redirect Pages, continued

“query” redirects

If, for some reason, the “include” method will not work for you (for instance, if the redirect page can not be in the same folder as ProcessForm.php), then you’ll want to use the “query” method. This redirect type uses a URL query string and JavaScript to print the form results into the redirect page.

This used to be the only method available, but I wasn’t happy with the limitations. Some of the possible problems are, if the visitor has JavaScript disabled (it doesn’t happen often, but it does happen), then they won’t be able to see the form results. Also, the form results are printed into the browser address bar, which might seem a bit sloppy. Another disadvantage is that some browsers have a limit on how much data can be passed in a URL query string, so if there are any very long text fields, some of the form data may be cut off in the confirmation page.

To use the “query” method, you’ll need to make sure you use a “redirect_type” field with the value set to “query”. Then, you’ll need to include a simple JavaScript in your redirect page where you’d like the form results to appear. The specific instructions for installing and using the JavaScript are included with the script in the “Extras” folder that was downloaded with ProcessForm and this PDF file...



CONTINUE >

Save Results to Text File

You can also have the ProcessForm script write the form results to a text file on your server (for archiving feedback and building mailing lists and such), and also include the form contents with the email as an attached text file (for easier importing into your own spreadsheet or database). Both options let you control the formatting of the text file and the separation characters.

attach_text_file

If you want to include the form information as an attached text file in your resulting email, just make a new hidden form named “attach_text_file” and set the value to be the name of the new file you’d like it to create and include as an attachment (best to use a .txt extension for the file name).

write_to_file

```
(moved to script configuration settings, line 91)
```

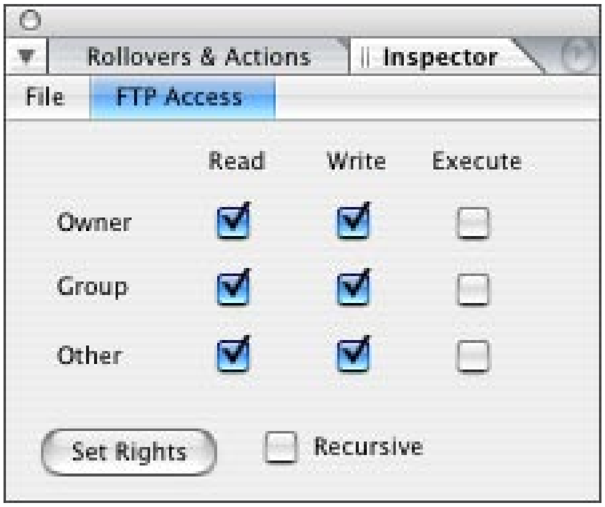
To have the script write your form contents to a text file on your server, add a new hidden field named “write_to_file” and set the value to be the URL address of the text file you want the results to be written to. It’s easiest to just put this file into the same folder as the script file and just enter the file name as the field value, but it’s not mandatory in this case.

force_format

```
<input type=”hidden” name=”force_format” value=”email, last_name, first_name”>
```

This optional function allows you to have only certain fields in your form be written to the text file. For instance, if your visitor is filling out a complicated profile form, but you just want the name and email address for a mailing list, then you’d make a new hidden field named “force_format” and set the value to be the list of form field names, in the order you’d like them to be, separated by commas.

If using a “write_to_file” field, we need to make a text file, upload it to the server, and give it the right permission settings. Create a new empty simple text document and save it with a .txt extension. Upload this file to the server. After it’s been uploaded, you’ll need to set the permission of the file. Many FTP applications have controls as shown to the right. If so, check the boxes to match.



	Read	Write	Execute
Owner	☑	☑	□
Group	☑	☑	□
Other	☑	☑	□

Otherwise, you’ll need to use whatever tools are available on your server to set the permissions to all read, all write and none execute...
chmod 666 (-rw-rw-rw-)

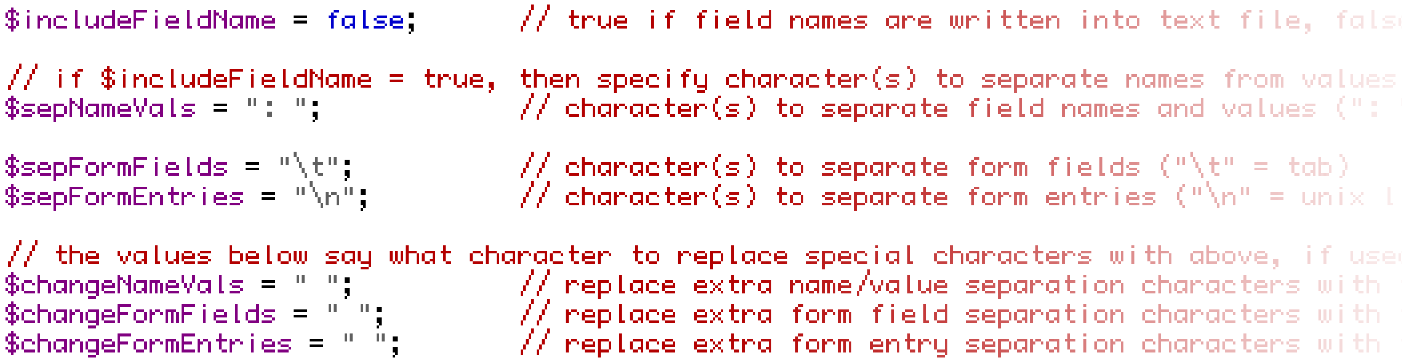
For better security, save this text file into your cgi-bin folder or any other folder above the web root directory on your server. Usually you’ll need to enter an absolute server path as the URL to a file above the web root (for the “write_to_file” field), unless there’s also a cgi-bin alias folder in your site root. You may need to contact your web host to get this information.

At this point I’d recommend testing things before continuing with the other options. The script should already have everything it needs to write the form results to the text file. It’s not likely, but there’s a possibility that you’re unable to change file permission settings on your server. If that’s the case, we probably won’t be able to use the text file features and it would be pointless to spend any more time with the details. Of course, if things don’t seem to be working right, make sure everything else is set up according to the instructions before giving up hope. If everything’s working at this point, then we’re ready to do some fine-tuning to get the text file to work just how we want.

CONTINUE >

Save Results to Text File, continued

There are quite a few settings you can make inside the ProcessForm.php file to change how it handles writing the form results to your text file. It’s simple to make these changes in the script, just be careful to use quotes around values where needed and not change anything other than the value unless you’re familiar with PHP. To get started, open the ProcessForm.php file and scroll down to around line 87 (this may vary depending on how many recipient lines you ended up using). By default, things should look about like this...



```
$includeFieldName = false;       // true if field names are written into text file, fals

// if $includeFieldName = true, then specify character(s) to separate names from values
$sepNameVals = ": ";             // character(s) to separate field names and values (":

$sepFormFields = "\t";           // character(s) to separate form fields ("\t" = tab)
$sepFormEntries = "\n";          // character(s) to separate form entries ("\n" = unix l

// the values below say what character to replace special characters with above, if use
$changeNameVals = " ";           // replace extra name/value separation characters with
$changeFormFields = " ";         // replace extra form field separation characters with
$changeFormEntries = " ";        // replace extra form entry separation characters with
```

$includeFieldName

This value should be either true or false, with no quotation marks (but make sure you don’t delete the semicolon). If you change this to true, it will write the field name into the text file in addition to the field values. By default, this is set to false so the text file is easily imported into a spreadsheet program such as Excel.

$sepNameVals

This variable would only be used if the $includeFieldName value above was set to true. This is the text character(s) placed between field names and value in the text file. By default, the name/value pairs would look like “Name: John Doe”. If you changed the setting to be “ - “, then it would look like “Name - John Doe”. Make sure you include any spaces where needed, and this value should always be inside the quotation marks.

$sepFormFields

This value determines what is used to separate the different form fields. By default, we’re using “\t”, which means a tab. This will rarely be changed.

$sepFormEntries

This value determines what is used to separate different form submissions in the text file. By default, we’re using “\n”, which means a line break or carriage return. If those line returns aren’t being recognized when opened in your local computer, you might try “\n\r”. This setting only applies to forms using the “write_to_file” field, and isn’t needed for the “attach_text_file” option.

$changeNameVals, $changeFormFields, $changeFormEntries

You aren’t likely to need to change these values, but they are there in case someone enters one of your special separation characters into the text field, which might corrupt your text file if you tried to import it into a spreadsheet or database. By default, we’re just using spaces. For instance, if we’re using tabs to separate form fields, and someone enters a tab into a form text area, then the script will change that tab to a space before writing it into the text file.

CONTINUE >

Save Results to MySQL Database

Another big new feature in version 3.0 is the ability to write your form results into a MySQL database. This tutorial assumes that you already have your database table built and ready to use on the server.

write_to_mysql

```
<input type="hidden" name="write_to_mysql" value="email, name > contact_name,
company, phone, fax, comments > user_comments">
```

Add a new form field named "write_to_mysql" and for the value, enter the names of the form fields that will be added to the database, separated by commas. If the database field name is the same as the form field names, then we're already set. If the database field names are different, though, put it beside the form field name with a > character between them (form field name to the left, database field name to the right), and still use commas to separate the pairs. See the illustration above - "email" is the name of the form field AND the database field, but for the next one, "name" is the form field name and will be written into the database field named "contact_name".

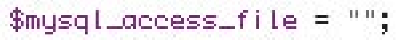
mysql_table

```
(moved to MySQL connection file, ProcessFormDB.php - see below)
```

Now we need to tell the script which database table to write the results to. Just add a new hidden field named "mysql_table" and set the value to be the name of the table in your MySQL database that will hold your form results.

Now we need to give the script all the information it needs to get permission from your server to access the database. Included with this download in the Extras / MySQL Utilities folder, you'll find instructions and another PHP file named ProcessFormDB.php - Open this file and make the database, hostname, username and password settings for your server. Then upload the file to a "safe" directory as specified in the ReadMe.txt file.

We now need to tell the main ProcessForm.php file where this script file is located on the server. Open ProcessForm.php again and on around line 84 enter the path to where you uploaded the ProcessFormDB.php file (enter the path between the double quotes after $mysql_access_file). You may need to set this path as an absolute path from the web server, so see the ProcessFormDB.php ReadMe.txt file for details.



```
$mysql_access_file = "";
```

updating database records

```
<input type="hidden" name="mysql_update_field" value="id">
```

```
<input type="hidden" name="mysql_update_value" value="1">
```

your form, then add 2 new hidden fields. Make one named "mysql_update_field" and set the value to the name of your database field that holds the unique record ID number. Make a second field named "mysql_update_value" and set the value to the ID number or value of the record to update. In this example, the record with the field "id" set to "1" would be updated.

A good security practice when writing to a MySQL database is to create a separate database user in MySQL that only has "INSERT" permissions (and "UPDATE" if needed). That will help keep unwelcome guests from selecting, editing or deleting anything in the database in case your server is compromised.

CONTINUE >

File Uploads / Attachments

Version 3.0 added another big feature - support for file uploads and attachments. You won't have to do anything to the script to enable this - it's ready to go by default - you'll just need to make sure your HTML form is equipped to handle file uploads.

Use the form file input tag so your visitors will be able to attach their files to your form...

```
<input type="file" name="fileGetterName" size="16">
```

Also, make sure the form encoding type (in the form tag) is set for "multipart/form-data"...

```
<form action="ProcessForm.php" method="post" enctype="multipart/form-data">
```

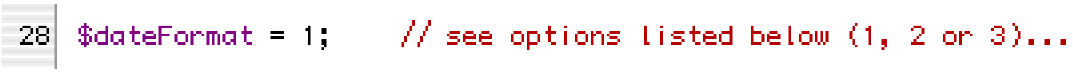
The default maximum size for file uploads in PHP is 2MB, so your form would be subject to those limitations. Your web host may have changed that setting from the default. Many email servers also have fairly low file size limitation. Basically, just don't expect your form to be able to handle files of unlimited size.

Script Configuration Options

Aside from the features we've already covered, there are additional configuration options that you can edit as needed inside the script file. These changes will affect all forms that are submitted to the script file and not just individual forms.

$dateFormat

Around line 28, you'll see an option for $dateFormat. The available values are either 1, 2 or 3 (not inside quotation marks). The default options is 1. These numbers correspond with a time/date pattern that will be used by the script.



```
28 $dateFormat = 1;     // see options listed below (1, 2 or 3)...
```

The patterns that each number uses are as follows:

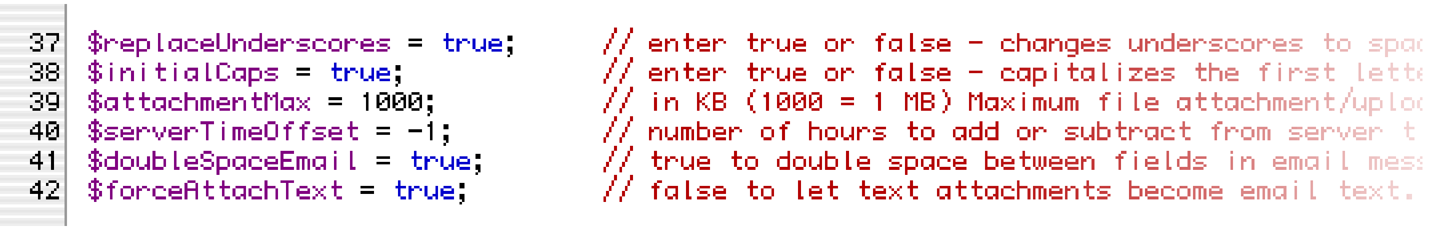
1: January 4, 2004 @ 6:00 pm
2: 2004-01-04 @ 18:00
3: 4 January 2004, 18:00

When the form has been processed, the current data and time will be written in one of the above patterns into the email message and the default confirmation page.

CONTINUE >

Script Configuration Options, continued

Lines 37 - 42 (depending on how many recipients you've added) are settings for an assortment of other options. These are explained in detail below.



```
37  $replaceUnderscores = true;      // enter true or false - changes underscores to spac
38  $initialCaps = true;             // enter true or false - capitalizes the first lette
39  $attachmentMax = 1000;           // in KB (1000 = 1 MB) Maximum file attachment/uploa
40  $serverTimeOffset = -1;          // number of hours to add or subtract from server t
41  $doubleSpaceEmail = true;        // true to double space between fields in email mess
42  $forceAttachText = true;         // false to let text attachments become email text.
```

$replaceUnderscores

When working with HTML forms, it's often a good idea to make sure the names of all your form fields contain only letters, numbers and underscores (no spaces or special characters). This is particularly important if you're also using JavaScript validation. While good coding, it doesn't necessarily look good in the email and confirmation page. If
$replaceUnderscores is set to true (no quotes), then the script will replace the underscores with spaces in your field names so they are easier to read. For instance, in the email and confirmation page, the field name "first_name" would be changed to "first name". To turn this option off, simply change it to false.

$initialCaps

This option also exists to make your email and confirmation page easier to read. If set to true (no quotes), it will capitalize the first letter of each word in your form field names. For instance, if used with $replaceUnderscores, "first_name" would become "First Name". To turn this option off, change the value to false.

$attachmentMax

This setting sets the maximum size of file uploads / attachments in KB. By default, this value is set to 1000 (1 MB). Whatever you decide to change this to, it is still dependent on your servers maximum file size for PHP, which is 2 MB by default.

$serverTimeOffset

This value exists in case you and your server are not in the same time zone. This number (no quotation marks) will adjust the time/date that's printed in the email and confirmation page by however many hours you enter. Negative numbers are also allowed. If your server says 5:00pm and you need 7:00pm, you'd enter 2 for the value. If your server says 5:00pm and you need it to say 2:00pm, you'd enter -3 for the value.

$doubleSpaceEmail

This option determines whether the results in your email message are single spaced or double spaced. By default it's set for double spacing, so change this value to false (no quotes) if you'd like it single spaced.

$forceAttachText

If your visitor attaches a plain text file (.txt) in a form that uses file attachment, this setting (true by default) forces the text to remain as an attachment. If you change this to false (no quotes), it will allow the text to be simply written into the email at the bottom.

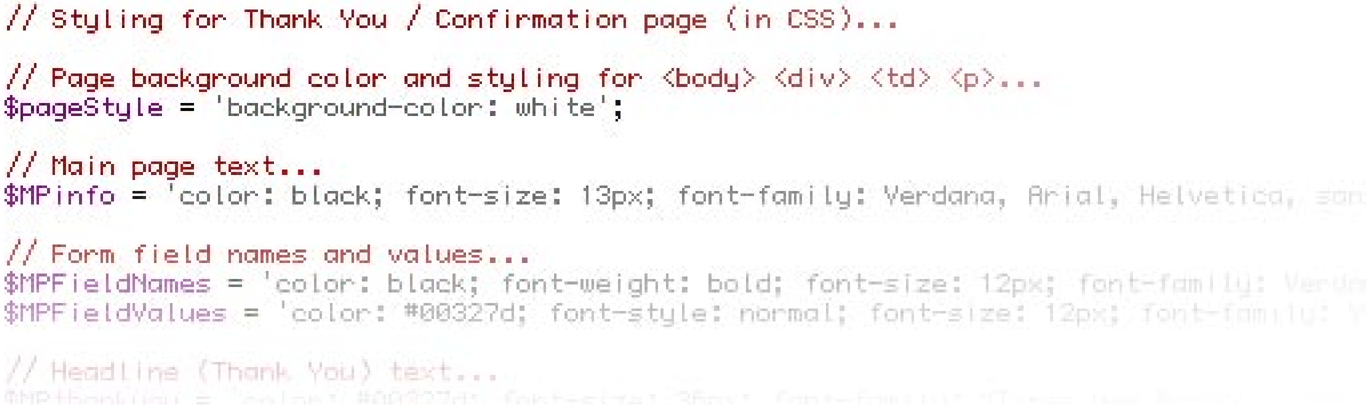
CONTINUE >

Confirmation Page Styling

The built-in confirmation page is highly customizable. The text that appears on this page and the error messages the visitor would see if there was a problem with what they entered is all customizable. These setting are, by default, in lines 102 - 156 of the script. As with any other script changes, I'd recommend making a back-up of your ProcessForm.php file just in case. Most values are between quotation marks, so if you use any quotation marks in any of your messages, make sure they're escaped with a backslash, otherwise you'll get a script error. For instance...

$fakeVariable = "The word \"hamster\" is surrounded by escaped quotation marks.";

If you'd like to change the background color of the page, or any of the font styles of the text, the CSS settings for the confirmation page are in lines 46 - 78. You do not need to "escape" quotation marks in these lines, but will need to escape any apostrophes or single quotes with a backslash. Otherwise, if you're familiar with CSS, these changes should be easy to make. Even if you're not, most of the language is self-explanatory, so you might just make a back-up and give it a shot. If you aren't comfortable with editing these settings, you might want to look into using a "redirect" page (see page 8).



```
// Styling for Thank You / Confirmation page (in CSS)...

// Page background color and styling for <body> <div> <td> <p>...
$pageStyle = 'background-color: white';

// Main page text...
$MPinfo = 'color: black; font-size: 13px; font-family: Verdana, Arial, Helvetica, san

// Form field names and values...
$MPFieldNames = 'color: black; font-weight: bold; font-size: 12px; font-family: Verda
$MPFieldValues = 'color: #00327d; font-style: normal; font-size: 12px; font-family: V

// Headline (Thank You) text...
$MPHeadline = 'color: #00327d; font-size: 36px; font-family:
```

Congratulations!

At this point, you're finished with all my ramblings and should pretty much have an idea of how things work. If you run into any problems, there are plenty of ways to get support. Don't forget about the "Test PHP Email" utility in the Extras folder that came with this download. Also, I'd encourage you to check over the Troubleshooting.pdf file that was also included with your download.

If your question has still not been answered, feel free to visit our support forum and post your questions online. I personally monitor that forum along with some other very knowledgeable people, and you're likely to get an answer sooner than sending a direct email...

www.mindpalette.com/forum

THE END :-)