

Flash-Based Email Form Using PHP

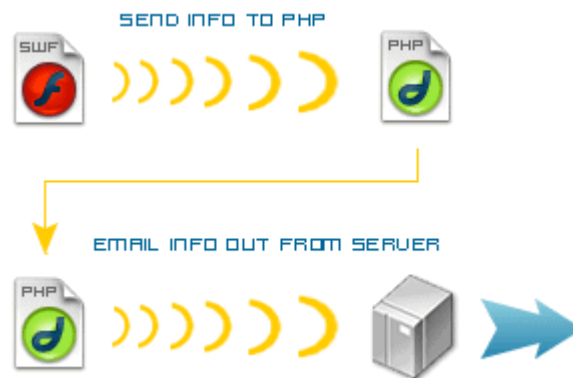
by SorBose, Inc.

Introduction

Here we will cover the process involved in making a very simple email form in Flash using PHP. Be aware that Flash alone is incapable of sending email. PHP, or some other server-side script like PHP, is needed to handle that operation. Put the two together, though, and you got yourself a great way for people to email you directly through your Flash-based web site.

How it Works

Because PHP is being used to send an email, Flash needs a way to communicate to PHP to let it know when to send an email and with what information. Flash relies on PHP to handle the email sending responsibility. When a Flash movie is ready to have an email sent, it needs to talk to a PHP script on the server so that script can then instruct the server to send the email with the information provided.

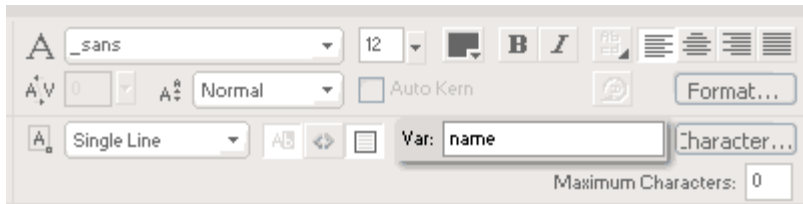


A way Flash can send information to an external PHP page is through the *loadVariables* Actionscript command. This allows Flash to open a page on the server, like a PHP script, and send it specific information that allows you to control how that script behaves. In the case of this tutorial, to have a PHP script email information inputted in a Flash form. Flash sends the PHP page the information filled out in the form. The PHP script then inserts that information into a *mail* function which instructs the server to send an email with that information to the email address specified. Easy enough, right? Lets begin creating the Flash form.

Steps to Create Form

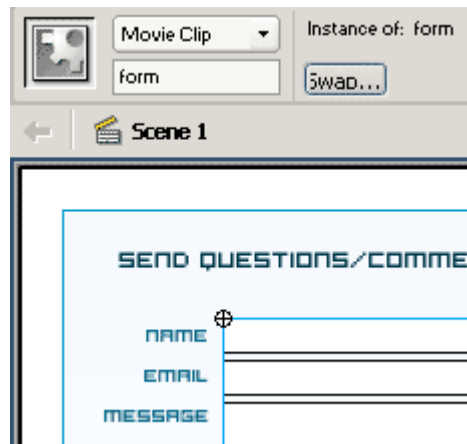
The Flash form consists of 2 basic parts. One part is the group of text fields that make up the form. These are actually contained within a movieclip aptly named "form." Secondly, you have the send button. This will be the button that activates the code that sends the form information to the PHP file. At that point, Flash's job is done. It's then up to the PHP script to make sure the email gets sent.

- i. Start off by making the appropriate form fields. Make sure these are input fields and not static or dynamic text fields Include as many as you want. Each will be sent to the PHP file where they can then be sorted. This example uses 3.
 - ii. Assign each text field a var value. This is NOT an instance name. The var field allows you to associate a variable with the given text field Because *loadVariables* is being used to transmit the information, this is needed to make the value of these text fields easily recognized as variables to that command. This example uses name, email, and body for field variable names.



[assign a var value for each text field]

- iii. Once you have created and named each field, select them all and create a new movieclip out of them. This will be the form movieclip. Give it the instance name form when you're done.



- iv. Next, create a button. This will serve as the send button. This will exist not within the form, but in the same place as the form. It's on this button that the loadVariables script will be added. That script is as follows:

```
form.loadVariables("email.php", "POST");
```

This calls loadVariables through the form movieclip sending all variables saved in that movieclip to email.php using the POST method. Because all the text fields in form have variables associated with them, this effectively sends all information filled out in those fields to the email PHP page. From that PHP page, the sent information can be retrieved using each field's var name. One thing to be cautious of is that you need to make sure the movieclip you are using loadVariables with exists long enough for it to send its variables to the URL specified. Because this mailer has a thank you screen after sending an email, you'll need to wait until Flash does that before showing that screen and losing the form movieclip.

- v. You can tell when a movieclip has sent its variables out from its onClipEvent(data) event. This is event is called when the movieclip receives data from the server either confirming its variables have been sent or in the case where new variables are brought into Flash. We aren't bringing variables in, just sending out. But we do need to know when that's taken place. So on the form movieclip we can add the script that checks for that. When the event runs, we can then go to the next frame showing the thank you screen.

```
onClipEvent(data){
```

```
    _root.nextFrame();  
}
```

Steps in Scripting PHP

Now we can write that PHP script, email.php, that really makes this happen. Luckily, it's not really all that difficult to write. Maybe not as easy as the previous Flash script, but not difficult nonetheless. The PHP all revolves around one function; the [mail](#) function. All you need to do is to grab the information sent from Flash and pass it in to this mail function in PHP to have that information sent to your email of choice.

- i. Create a PHP file if you haven't already. If you're not sure how, just make a text file. A PHP file is basically nothing more than a text file with a .php extension instead of .txt. To a server, though, a php file is seen not as text, but as a script for carrying out commands - commands like our email command.
 - ii. Using Dreamweaver or your text editor of choice, start writing your PHP script. This will consist of 3 basic parts. One will be a place for constant variables that will always remain the same. This will include things such as your email. Another is the capturing of those variables sent to the script from Flash - those filled out in the Flash form. Finally, that information is then setup in the third part, the call of the mail function.
 - iii. Part 1: Begin the file with a php tag and the setting up of constant variables. This includes your email and most likely a subject line as well.

```
<?php  
$sendTo = "senocular@hotmail.com";  
$subject = "My Flash site reply";
```

- iv. Part 2: Now you can start getting the variables that Flash sent. Since Flash sent the variables using post, we would use \$_POST to get them into variables of our own. \$_POST is a special global variable in a PHP script that contains all of the posted variables sent to that script as an associative array. Using brackets ([]) and a variable name, you can then retrieve those variables. For those variables sent by this particular example, you would obtain their value in PHP using \$_POST["name"], \$_POST["email"], and \$_POST["message"].

```
$headers = "From: " . $_POST["name"];  
$headers .= "<" . $_POST["email"] . ">\r\n";  
$headers .= "Reply-To: " . $_POST["email"] . "\r\n";  
$headers .= "Return-Path: " . $_POST["email"];  
$message = $_POST["message"];
```

*Name and email here are assigned to a single variable, headers, because of how the mail function works...

- v. Part 3: Finally the mail function. This function is set up to take 3 to 5 different parameters. We're only concerned about 4:

```
mail(recipient, subject, message, other headers);
```

With our previous variable definitions, we can easily just plug and play to run this command in the final php script. Note that "from" and "reply-to" are not separate parameters within the mail call. These are part of the 4th parameter, additional headers, where information beyond the recipient, subject and email message go. With that, we can finish off the script with the following.

```
mail($sendTo, $subject, $message, $headers);  
?>
```

vi. Save as email.php and you're set.

Putting it All Together

All that remains now is uploading your published swf (with accompanying html) and PHP file to your server. Be sure you keep the PHP file in the same directory as your swf and html or your script may not be found when called from Flash. If you don't want it to be in that same directory, be sure you correctly reference the location of the PHP file in the loadVariables command used in Flash. Once uploaded, play your movie and send yourself a message! It'll be fun - and it will also test to make sure this actually works ;) since, if you didn't already know, you can't test from your own computer's hard drive unless you have the proper configurations and PHP installed (which, more than likely, is not the case). It's ok though, since you don't need it on your personal computer, just your server. And if its working from there, then you should be in the clear.