

# Sending Mail Using Flash 4+ and PHP

In another tutorial (or perhaps just on your own!), you learned that sending form data in e-mail is very simple with PHP. Now you can pretty-up that boring ol' HTML form and use a Flash 4 movie as the front-end to the PHP script.

NOTE: This tutorial uses the PHP file extension ".phtml" in its examples. If you use a different file extension to indicate PHP files, please substitute accordingly. Also, you must have a valid value set in the SMTP section of the `php[3].ini` file.

There's no PHP trick to this process. In fact, there's no trick at all - the Flash movie points to a script that sends the mail, and that script could be a PHP script, a Perl script, or any other server-side scripting language that you want to use. In this first tutorial in a series focusing on Flash/PHP interaction, the goal is for you to become familiar with the most basic of concepts: text field names and how those variables are passed to the script.

This tutorial assumes you are familiar with creating Flash 4+ movies. If you're not, please visit the Macromedia web site to download an evaluation copy of Flash and its accompanying "Getting Started" help files. Otherwise, please feel free to download the source file package for this tutorial, which includes:

- **flashy\_mail fla** - the Flash source file
- **flashy\_mail.swf** - the exported Flash 4 movie
- **show\_flash\_form.html** - the HTML file that loads the Flash movie
- **send\_flash\_form.phtml** - the PHP script that sends the mail and displays a confirmation.

The important parts of the Flash movie are the text fields and the button action. In this example, we have three text fields: one to hold the user's name, one to hold the user's e-mail address, and one to hold the message. The name of the text box in Flash is similar to the use of the NAME attribute in an HTML form input field: this will be the name of the variable sent to the script.

After you have created text fields in Flash, assign properties to them, such as name, multiline and word wrap (or no word wrap). The figures below show the properties of the fields in our example movie:

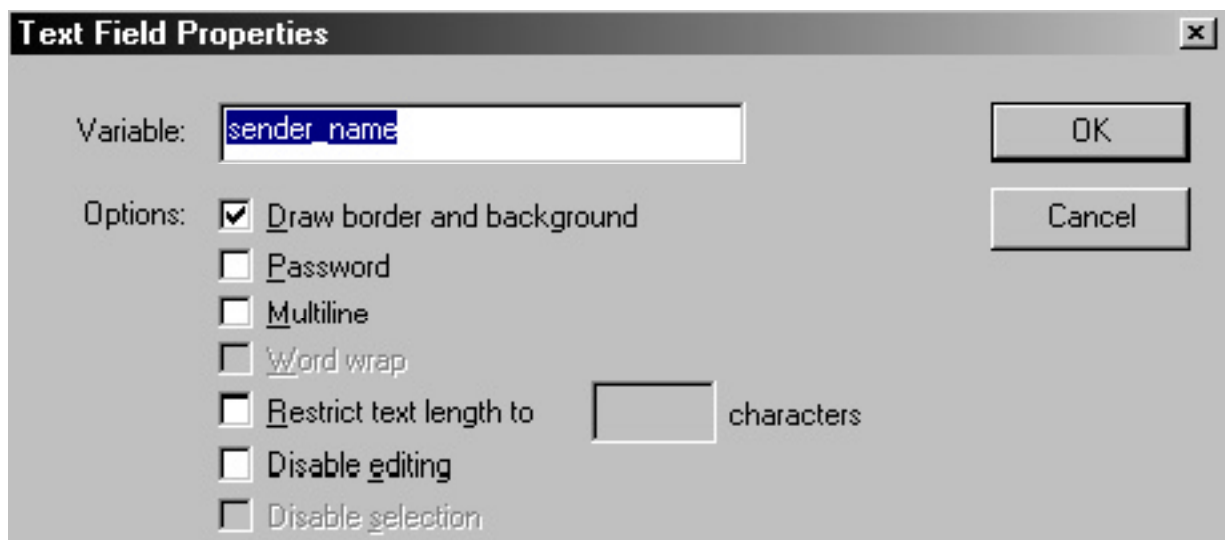


Fig 1. "Your Name" field properties

## Sending Mail Using Flash 4+ and PHP (cont'd)

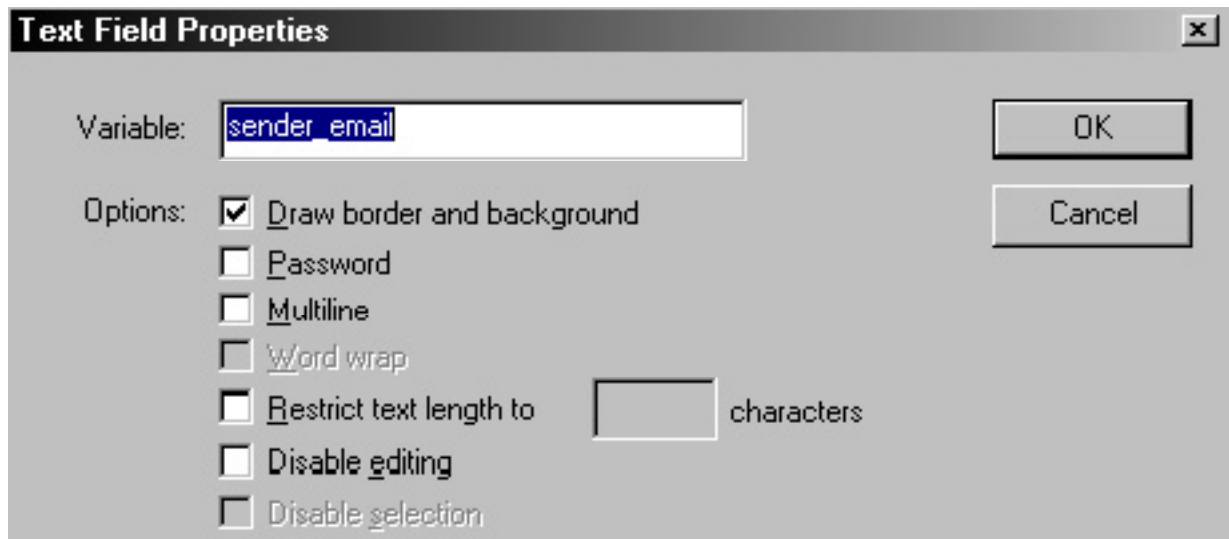


Fig 2. "Your E-Mail" field properties

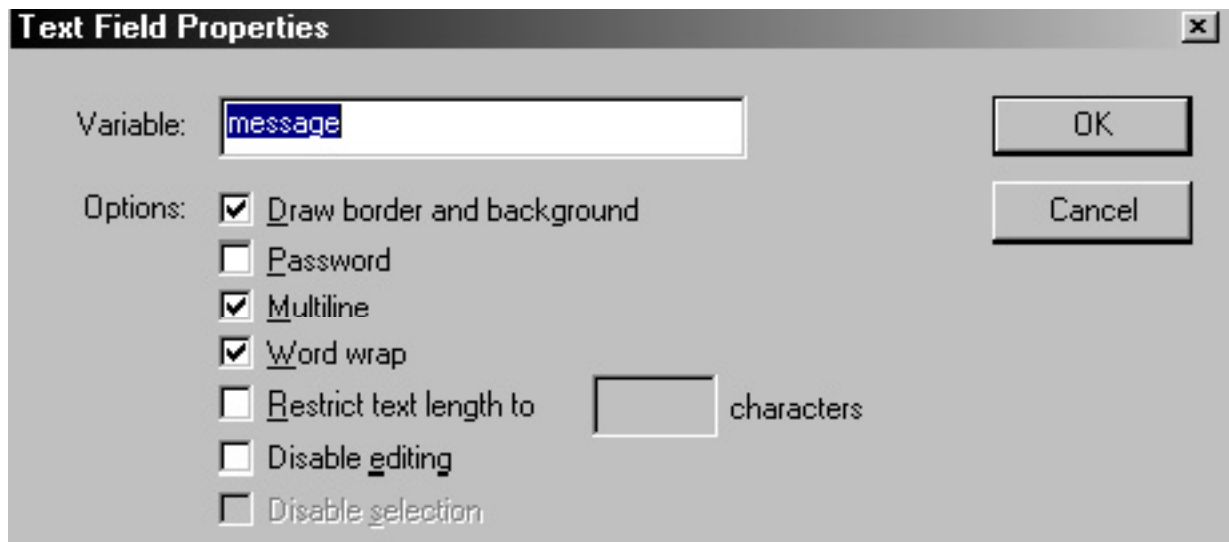


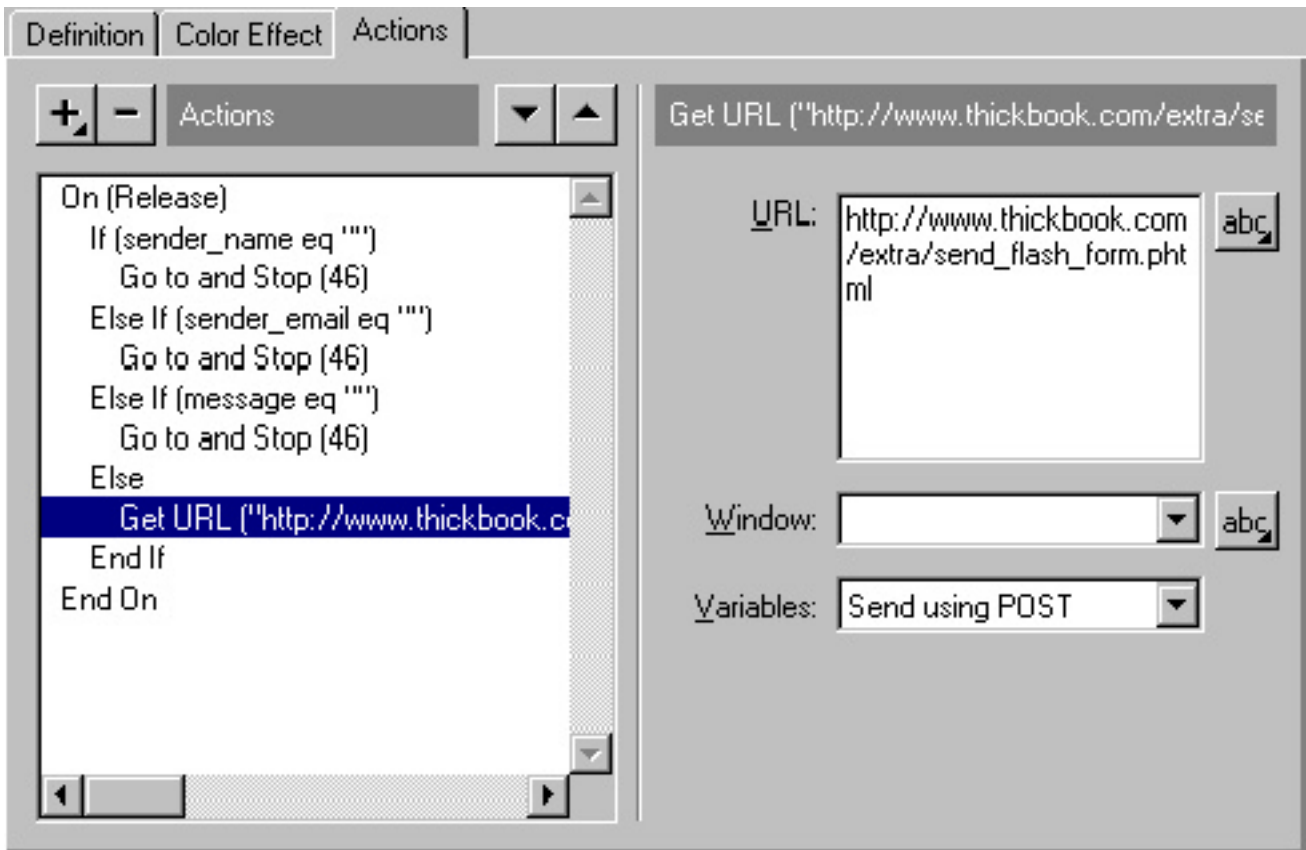
Fig 3. "Your Message" field properties

Note the use of the "multiline" and "word wrap" for the message field. Additionally, it's up to you to draw the text field as large as you would like it to be. While HTML has the "cols" and "rows" attributes for textareas, you actually have to draw a large enough area within your movie.

Now you have three fields: "sender\_name", "sender\_email" and "message", ready for user input. Next, we make it send...

Create a button of some type and assign a series of actions to it, which will execute when a particular mouse event occurs. The figure below shows the script assigned to the button in our example form, which I'll explain step-by-step in a moment.

## Sending Mail Using Flash 4+ and PHP (cont'd)



**Fig 4. Button Action**

You'll see that the language of Flash movies is a relative of most scripting languages, with its if...else if...else construct and string comparison operators. The overall action in this case is the "On (Release)" action. This means that the code within the "On" block is executed when the button is released (after it is pressed).

The if...else if...else block is the validation routine. It takes the name of each of the text fields and looks for the presence of any characters. If no characters are found (i.e. the string is empty), the action within the construct is executed. Otherwise, the loop continues, and executes the final action.

So what exactly does "Go to and Stop(46)" mean, and why is it executed when a string is empty? Well, the movie is made up of frames, and the number of frames in this movie totals 46. However, for all intents and purposes, the movie stops playing at frame 45. Frame 46 only becomes visible if one of the validations is true, such as an empty text field for "sender\_name". The content of frame 46 is a piece of red text that says "You missed a required field!". Therefore, if you press the "Send Mail" button and any of the three fields are empty, you'll see frame 46, and you'll always see frame 46 until all three fields are filled in and the form can be sent.

Now, the fun part: getting the form to send! The last part of the button's if...else if...else action is "Get URL ("http://www.thickbook.com/extra/send\_flash\_form.phtml", vars=POST)". Don't be fooled by the use of the term "Get", because you can either GET or POST variables with "Get URL". This portion of the Flash script is essentially the action of the form. If you were to write this in HTML, it would be:

```
<FORM method="POST" action="http://www.thickbook.com/extra/send_flash_form.phtml">
```

## Sending Mail Using Flash 4+ and PHP (cont'd)

So, now all you have to do is create the simple PHP script to handle the form input. Start by adding some validation to the script. Just because you have it within your Flash movie doesn't mean that you shouldn't have it in your script as well!

Start your PHP script with a statement such as the following, which redirects a user to the form if the required fields aren't complete. Substitute your own URL, of course.

```
if (($sender_email == "") || ($sender_name == "") || ($message == "")) {  
    header("Location: http://www.thickbook.com/extra/show_flash_form.html");  
    exit;  
}
```

Next, create a variable to hold the value of the mail recipient, for example:

```
$recipient = "julie@thickbook.com";
```

Now, a variable to hold the subject, for example:

```
$subject = "Flashy Mail From thickbook.com";
```

Add some mail header information:

```
$mailheaders = "From: <$sender_email> \n";  
$mailheaders .= "Reply-To: <$sender_email>\n\n";
```

Now, build the message string:

```
$msg = "Sender's Name:  $sender_name\n";  
$msg .= "Sender's E-Mail: $sender_email\n";  
$msg .= "Message:      $message\n\n";
```

Finally, put it all together within the mail() function:

```
mail($recipient, $subject, $msg, $mailheaders) or die ("Couldn't send mail!");
```

Add in some HTML to echo back the contents of the mail, and your entire script should look something like this:

```
<?php  
if (($sender_email == "") || ($sender_name == "") || ($message == "")) {  
    header("Location: http://www.thickbook.com/extra/show_flash_form.html");  
    exit;  
}  
  
$recipient = "julie@thickbook.com";  
$subject = "Flashy Mail From thickbook.com";  
  
$mailheaders = "From: <$sender_email> \n";  
$mailheaders .= "Reply-To: <$sender_email>\n\n";
```

## Sending Mail Using Flash 4+ and PHP (cont'd)

```
$msg = "Sender's Name:  $sender_name\n";
$msg .= "Sender's E-Mail: $sender_email\n";
$msg .= "Message:      $message\n\n";

mail($recipient, $subject, $msg, $mailheaders) or die ("Couldn't send mail!");
?>

<HTML>
<HEAD>
<TITLE>Form Sent!</TITLE>
</HEAD>
<BODY BGCOLOR="#FFFFFF" TEXT="#000000">
<div align=center>
<p class=h2>Form Sent!</p>
<P>Thanks, <?php echo "$sender_name"; ?> (<?php echo "$sender_email"; ?>),
for sending this message:</p>
<P><?php echo "$message"; ?></p>
</div>
</body>
</html>
```

That's really all there is to it: the mail gets sent, the user sees a confirmation, and all is right with the world. The next tutorial in this series shows you the reverse situation: sending variables from the PHP script to the Flash movie.