

Flash 4 and PHP - Retrieving Text From a Database

In the first Flash 4/PHP Tutorial, Sending Mail Using Flash 4 and PHP, we sent some variables from a Flash 4 movie to a PHP script. In this lesson, we'll go the other way: sending variables from a PHP script to a Flash 4 movie.

In this example, a MySQL database table contains a bunch (21, actually) of random "pearls of wisdom" that I grabbed from some random-quote-generator. The Flash 4 movie contains a button that, when pressed, executes a PHP script. This script does the following:

- queries the database to count how many quotes exist
 - picks a random number between 1 and the total number of records
 - retrieves the quote whose ID matches the random number
 - encodes the results using the `rawurlencode()` function
 - returns the string to the movie
- 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 4 and its accompanying "Getting Started" help files. Otherwise, please feel free to download the source file package for this tutorial, which includes:
- `random_quote fla` - the Flash 4 source file
 - `random_quote swf` - the exported Flash 4 movie
 - `random_quote.html` - the HTML file that loads the Flash 4 movie
 - `get_random.phtml` - the PHP script that gets a random quote from the database.
 - `db_schema.txt` - a text file containing the table creation and record insertion SQL statements
- The two most important parts of this Flash 4 movie are the text field, which will house the generated quote, and the button action, which executes the script. The name of the text field in Flash 4 is similar to the use of the NAME attribute in an HTML form input field. However, unlike a form that sends variables, in this case we're populating the field. It sounds kind of backwards, but trust me on this... After you have created the text field to hold the quote, assign the following properties to it: give it a name, make it multi-line and allow it to word wrap. Also, check the "disable editing" box. The figure below shows the properties of our "quote" text field:

Fig 1. "Quote" field properties Next, 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 quote-generator:

Fig 2. Button Action 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 action associated with this button is "Go to and Stop(32)", and the "32" refers to the frame number that contains the "quote" text box. In case you haven't guessed by now, the Flash 4 part of this equation is pretty simple. One more step and then we get into the PHP script that makes it "go". The action properties of the text field called "quote" are shown in the figure below:

Fig 3. Action Associated with "Quote" field

The action is "Load Variables ("http://www.thickbook.com/extra/get_random.phtml", 0)". It basically means, "run the script called `get_random.phtml` and whatever it returns, consider it to be a stream of variables and load it into the movie." So, knowing that's all there is to setting up the movie, let's jump into the PHP script and make it spit out some variables! A Flash 4 movie expects variables that are "encoded", meaning that all non-alphanumeric characters are replaced with a percent sign followed by two hex digits, such as "%20" for a space, "%3A" for a colon (":") and so forth. The PHP `rawurlencode()` function does this job quite nicely, but we'll get to that in a moment. (Actually, you can see the url-encoded string if you just access the "get_random.phtml" script directly using the URL above. Its ugly, though.) Since the text field is called "quote", we have to send the Flash 4 movie a string that starts with "quote=", where everything after the "=" is the text to place in the field. The following script example shows you just how we go about this. Start your PHP script by connecting to MySQL (or your database server of choice) and selecting the database to use: `$db = mysql_connect("servername", "username", "password")` or die ("Couldn't connect.");

```
mysql_select_db("yourDB", $db) or die ("Couldn't select db.");
```

Next, get a total count of records in the table. This number will be used as the max level in the `rand()` function:

```
$get_count = "select count(id) from flash_random";
```

```
$count_result = mysql_query($get_count)
    or die ("Couldn't get count.");
```

```
$count = mysql_result($count_result, 0, "count(id)");
```

Time to seed the random number generator, and generate the random ID:

```
srand(((double)microtime()*1000000));
```

```
$random_id = rand(1, $count);
Make another query of the database, this time getting the text of the quote whose ID matches the random ID chosen by
the rand() function: $get_quote = "select quote from flash_random where id = '$random_id'";

$quote_result = mysql_query($get_quote)
    or die ("Couldn't get quote.");

$quote = mysql_result($quote_result, 0, "quote");
Now, build and echo the encoded string. The following code snippet encodes everything after the "quote=" directive. If
the entire $send_quote string were encoded, the "=" would be encoded as well, and we can't have that: $send_quote =
"quote=";

$send_quote .= rawurlencode($quote);

echo "$send_quote"; From start to finish, the script looks like this: <?

// connect to db
$db = mysql_connect("servername", "username", "password")
    or die ("Couldn't connect.");

mysql_select_db("yourDB", $db) or die ("Couldn't select db.");

// get count of quotes, as high limit for rand()
$get_count = "select count(id) from flash_random";

$count_result = mysql_query($get_count)
    or die ("Couldn't get count.");

$count = mysql_result($count_result, 0, "count(id)");

// seed random number generator
srand((double)microtime()*1000000);

// generate random id with high limit if $count
$random_id = rand(1, $count);

// get quote based on $random_id
$get_quote = "select quote from flash_random where id = '$random_id'";

$quote_result = mysql_query($get_quote)
    or die ("Couldn't get quote.");

$quote = mysql_result($quote_result, 0, "quote");

// encode and echo quote
$send_quote = "quote=";

$send_quote .= rawurlencode($quote);

echo "$send_quote";
```

?> That's really all there is to it: the Flash movie field called "quote" gets populated with the value of "quote" extracted from the encoded string. The next tutorial in this series will show you how to load a series of variables (news headlines) from a database query, then make them clickable so that the text can be extracted from the database on-demand.