

How to Make Flash & PHP Talk

Author: [Joseph Miller](#) | Website: <http://www.indivision.net> |

Page 1



[1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [next»](#)

How to Send Variables Between Flash and PHP

I had to learn a lot of this on my own via trial and error since there was little available on the net that adequately explained how it works. I'm hoping this guide will save someone the hours of hammering that I spent. Once you get a few fundamental things down, I think you will find, as I have, that a flash/PHP combination is easy to set up and offers a lot of flexibility. I included a very simple flash movie and script to go along with this guide. I recommend downloading and briefly going over all of the files before reading on.

How It Works

Loading variables from a PHP script is very similar to loading from a text file. The main difference is that the script performs work on a separate timeline. You can call a PHP script from a flash movie by using the `LoadVariablesNum()` command like so:

```
loadVariablesNum ("scriptA.php", 0, "POST");
```

Remember that your flash movie and the script are operating on separate timelines. The flash movie doesn't just stop when you make the `LoadVariablesNum()` command. Instead, it activates the script and keeps moving. When the script is finished operating, the data that has been printed by the script is read by the flash movie.

Loading Loop

Because the script is on a separate timeline, you need to create a loop in your flash movie that waits for the script to finish. Here is the code that I used in the sample .fla to make the loop:

```
if (loading eq "YES") {
    gotoAndPlay (_currentframe-1);
} else {
    gotoAndStop (1);
}
```

When you click on the button that loads the script, the sample .fla also sets the 'loading' variable to 'YES'. The movie knows when the script is done because it tells the movie that 'loading=NO'.

Setting POST

You may have scripts that do not need variables from the flash movie. In that case, you can leave the `LoadVariablesNum()` option set to 'Don't Send Variables'. Otherwise, you will need to set this option to 'POST' if you want variables to be sent into the script. Keep in mind that only the variables from the movieclip that contains the `LoadVariablesNum()` command will be sent!

Variables In PHP

Variables in a PHP script have a '\$' in front of them. For example, the variable named 'who' in flash will be '\$who' in the PHP script.

The 'Print' Statement

The data that the flash movie will eventually read is placed using 'print' statements. For an example of how print statements are used, refer to the file named: `scriptA.php`. Also note that all print statements in the script will be read at the same time. It's easy to make the mistake of thinking that you can control the actions of the flash movie according to the placement of a print statement. For example, in the sample files, I have it set up so that the 'loading' variable is set to 'NO' at the end of the script. If the script contained more commands after this print statement, they would also be included in the data that gets sent to the flash movie. In addition, if I put the line at the beginning of the script, it would really make no apparent difference to the output either.

Be sure to format your print statements so that the separate variables won't get 'smashed' together. Imagine that the results of the script will most likely look like one, continuous line. For example, if your script includes:

```
Print "dogs=fun";
Print "cats=boring";
```

The resulting data that the flash movie will read will look like:

```
dogs=funcats=boring
```

Because the movie won't be able to distinguish where variables start and end, you will end up with undesirable results. To make it work, you must include '&' between statements. For example, you might arrange the above data like this:

```
Print "&dogs=fun";
Print "&cats=boring";
Print "&fish=food";
```

The resulting data will now look like this:

```
&dogs=fun&cats=boring&fish=food
```

The '&' between the variable statements allows flash to read them each as separate statements.

Final Notes

I tried to narrow the information in this tutorial down to the things that caught me up the most. If you need to learn more about PHP, [PHP.net](#) and [DevShed](#) are great places to start. I will also answer questions posted on the forum at [INDIVISION](#). Good luck and I hope this helps.