Switchbox

By Vic Fryzel – PHPBuilder.com

Whether you're new to programming, or an experienced developer, the fact remains true that application architecture is imperative to a good application. By good application I mean one that is fast, efficient, powerful, programmer-friendly, and reusable. These things make up the ideal application.

I've developed an application architecture called SwitchBox. It's somewhat derived from FuseBox, but far less cryptic, far faster, and in most cases, equally as powerful. SwitchBox when used properly can help to achieve the ideal application, because it is small, easy to use, fast, and powerful. Ok so let's start looking at SwitchBox.

SwitchBox works very simply. Generally, all files that do not display output to users, but rather compute data to later be displayed and included on output pages, are kept in their own directory. For our purposes, this directory will be called 'includes'. Now suppose we had a bunch of files in includes. It doesn't make sense to include all these files on every page does it? What if we just needed one or two files? If we included every file all the time, we'd just be parsing a bunch of files for no reason, and thus, making the server-load skyrocket. And what if all your pages need a variable, but you have to keep it in one file that that page doesn't really need, because redefining it in every page would be inefficient, and including that file for simply one reason would be inefficient as well. This is a bad thing, because in an ideal application there is efficiency right? So, let's just include only the files we need on each page, and put the variables that every page needs in an included file that every page needs. We're going to do this by creating one file. We will name this file 'SwitchBox.php'. SwitchBox.php will take care of all of our includes for us, that's it, just one file. This file will be included in all pages that need includes First I'll display the code of SwitchBox.php, then I will explain how it works.

```
<?php
// SwitchBox.php

// Define global variables used by all pages
$url_prefix = "http://www.myserver.com/";

// Create the SwitchBox
switch ($switch) {
    case "index" :
        require './include/fns_database.php';
        require './include/fns_index.php';
        break;

    case "catalog" :
        require './include/fns_database.php';
        require './include/fns_catalog.php';
        break;

    case "checkout" :
        require './include/fns_database.php';
        require './include/fns_checkout.php';
        require './include/fns_email.php';
        break;
```

```
    // Leave default empty incase we just need global variables and
no includes
    default:
        break;
}
?>
```

If you're new to PHP or programming, that was probably very confusing to you, but it will get easier as you read on. First, we declare the global variables that every page needs, in this case, one of those variables would be $url_prefix. By doing this, every page that includes SwitchBox.php has access to those variables. Kind've convenient right?

Next we create the SwitchBox. The SwitchBox works around the value of one variable, $switch. Depending on what $switch is defined as, the SwitchBox does something different. For instance, if $switch was 'index', then the SwitchBox would include all the files required to make index.php work. And if $switch was 'catalog', the SwitchBox would include all the files required to make catalog.php work. And if $switch did not equal anything, then the SwitchBox would not do anything. We would not want our SwitchBox to do anything if we just needed the global variables, and no files. Ok so you to setup includes for a new page, all you have to do is create a new case, and include or require the files needed by that page, then break out of the switch.

Now let's move on and find out how to use our SwitchBox.

Using our SwitchBox is even easier as it only takes two lines of code to implement. Let's look at the code of index.php, and then I will explain it.

```
<?php
// Set switch
$switch = "index";

// Call SwitchBox
require './includes/SwitchBox.php';

/* Follow with index.php's code */
?>
```

What's happening is simple. We set our switch (used by the SwitchBox so that the SwitchBox knows what to do), and then we call the SwitchBox. Depending on what $switch was, our SwitchBox takes the appropriate actions. In this case, our SwitchBox included the files 'fns_database.php' and 'fns_index.php'. After those two lines of code, the global variables in the SwitchBox, everything in fns_database.php, and everything in fns_index.php, are all available right on index.php. If you did not want to include any files for the page, you could simply leave the line:

$switch = "...";

out of the page. Then only the global variables would be available in the application.

Easy right? Ok so let's review...

We're trying to achieve an ideal application that's efficient, powerful, fast, reusable, etc. In order to achieve this, we had to put in five minutes of extra work, that will save us hours of time later. To add a new page to our application, we add a switch to the SwitchBox, and two lines of code to the page we're using the SwitchBox on.

The SwitchBox saved us processor load by only including files needed by each page, and in doing so

also increased page loading speed. It also made our code extremely reusable. And even still, our SwitchBox has made updating our page far easier, as we have to enter one line of code into our SwitchBox to achieve a new include. The SwitchBox also acts as a plan for our application. If we're curious as to what files a page uses, or why a page is saying a function is not defined, we can simply view the SwitchBox, and it shows right there what files are being used or not used by the page.

There you have it. I hope that this article has helped everyone to better achieve their ideal applications.

Vic Fryzel

****

Date: 04/20/03 15:51
By: Rob
Subject: Another suggestion

If you like this method, here's another possible way to do it.

Instead of naming the switch like "index" and then manually putting a line like this in:

```
$switch = 'index';

You can use $_SERVER['PHP_SELF'] and set that up as your switch
condition:

switch ($_SERVER['PHP_SELF']) {
case '/index.php':
require_once('file/to/include.php');
require_once('file/to/also_include.php');

}
```

Then, on your pages where you would normally put $switch = 'index' just take it out and make sure that the switch gets parse and ran.