

## Parsing a Syndicated News Feed Using PHP by Peter deHaan

I converted a couple of my sites from HTML or ColdFusion to PHP. Along the way, I managed to hone my lacking PHP skills and pick up a few useful tricks and snippets.

One of the first things I tried when I started using PHP was to parse an XML packet, which conveniently happened to be a blog. In this tutorial you will learn how to find some very useful PHP classes, and learn how to parse some Tutorials from our site.

Obviously, in order to proceed, you need to install PHP, or have it accessible through your hosting company. Our hosting company has PHP 4.3.4 installed, so note that it's the version we have tested the following examples on. Instead of having to reinvent the wheel, this example teaches you how to find and install some PHP classes from <http://pear.php.net/>. PEAR, short for "PHP Extension and Application Repository", is a "framework and distribution system for reusable PHP components". So, in short, PEAR is a place you can download pre-made modules for PHP. In my experience, this saved me hours of time because I could take advantage of other people's hard work instead of starting applications from scratch.

Anyways, on with the tutorial.

1. Launch a web browser and open <http://pear.php.net/>
2. Type "RSS" into the search field and submit the form.

You should be taken to the "XML\_RSS" package where you can view documentation, see which packages are dependant on this package, and view the changelog/license, etc. We're not interested in any of that, so simply click the "Download Latest" link to snag the most recent copy of the package. At the time of writing the most recent version happens to be Version 0.9.2.

3. Save the package somewhere on your hard drive, such as your desktop.

The file is compressed, so if you don't have any sort of decompression utility, go to <http://www.rarsoft.com/download.htm> and download a trial version of WinRAR. After you've successfully decompressed the files and saved them on your local machine, you should see that there are a couple files in the "tests" folder.

4. Open the 01-parse.phpt file in Notepad and copy the following lines to the clipboard:

```
<?php
require "../RSS.php";

$r =& new XML_RSS("test.rss");
$r->parse();

foreach ($r->getItems() as $value) {
    echo $value['title'] . ": " . $value['link'] . "\n";
}

?>
```

5. Paste these few lines into a new file called parse.php in the same folder as 01-parse.phpt. Before you close the parse.php file, change the first line of code to the following:

```
require "classes/RSS.php";
```

6. Next, change the following line from:

```
echo $value['title'] . ": " . $value['link'] . "\n";
```

to:

```
echo $value['title'] . ": " . $value['link'] . "<br />\n";
```

The "." is what is known as a "concatenation operator"; or in laymans terms it joins strings. This is PHP's equivalent to the "+" symbol in Flash, or the "&" in ColdFusion. In either of these languages the string on the left hand side of the "." is joined with the string on the right.

7. Save the file and time to transfer the files to your PHP enabled web server.

For the sake of cleanliness, I uploaded both the parse.php and test.rss files into a folder named "rss" in the root of our website. In the new "rss" folder I also created a subfolder called "classes" where I uploaded the RSS.php file.

8. With all these files uploaded to your server, point your browser to the parse.php page. If everything went well, you should see the following output in your browser window:

```
PHP homepage: http://php.net/  
PEAR homepage: http://pear.php.net/  
PHP-GTK homepage: http://gtk.php.net/  
PHP QA homepage: http://qa.php.net/
```

Success!

Now, as borderline useful as that is, let's change the URL to grab something a little more interesting.

1. Replace the "test.rss" with the Flash-MX tutorials RSS file, which you can find at [www.flash-mx.com/tutorials/index.rdf](http://www.flash-mx.com/tutorials/index.rdf).
2. If you reload the page in your browser you should now see our 10 most recent tutorials posted to the site. Although the formatting leaves a lot to be desired, modifying the code is a fairly simple task.
3. Again, modify the line with the "echo" statement to the following:

```
echo "<a href=\"\" . $value['link'] . \"\" target=\"_blank\">\"  
    . $value['title'] . "<br />\n";
```

OK, before I explain the code, save the modified parse.php file, upload it to your web server and refresh the page in your browser. If all is fair in this world, the tutorial names should now be hyperlinked and only the titles display. If you click on a link it launches the tutorial in a new browser window.

So, now that you've seen what the code looks like in your browser, what exactly is the code doing? As with the previous example, the "echo" statement sends text to the browser window. In this case, it says "send a list of hyperlinked tutorial name to the user's browser".

The next logical question is "why do we need all those backslashes (\)?" Because the echo statement is enclosed in a pair of double quotes, you need to "escape" the actual quotes that you want sent to the browser by prefixing them with a single backslash. So in this example you want

quotes to appear in our source code around the both the href and target property in the source code. Naturally you can "view source" in your browser window to see what the final rendered output sent to the user's browser looks like. You should see something similar to the following:

```
<a href="http://flash-mx.com/tutorials/archives/000154.php" target="_blank">Using
_lockroot in Flash MX 2004.</a><br />
<a href="http://flash-mx.com/tutorials/archives/000153.php" target="_blank">Using Local
Shared Objects in Flash MX: The Flash Cookies</a><br />
...
<a href="http://flash-mx.com/tutorials/archives/000123.php" target="_blank">Displaying
Forta's "Tip of the Day" using Flash Remoting</a><br />
```

So, each link is on its own separate line, and followed by a `<br />` tag. The other thing you may have noticed is the `"\n"` at the end of the echo statement. In PHP, `\n` represents a newline character. This means a linefeed appears in your source code.

There are several other ways to send content to the screen apart from using "echo". You could also use "print" or "printf" which can sometimes make your source code slightly more readable. An example of using the printf can be seen below, remove the line containing the echo statement and replace it with the following:

```
printf("<a href=\"%s\" target=\"_blank\">%s</a><br />\n", $value['link'],
$value['title']);
```

Now, instead of using the concatenation operator (that pesky "." symbol), you define a placeholder for a string using "%s". Not a huge improvement, but it does make the HTML source code a bit easier on the eyes. You can see that there are two places in the above printf statement where you've used %s, in the href property and between the opening and closing tag. The two values at the end of the printf statement correspond to the two instances of %s defined earlier. The first %s will be replaced by the first parameter, or in this case \$value["link"]. The second %s is replaced by \$value["title"]. You could define as many parameters as you want, as long as they each have a matching %s (there are several other types you can use as well, as outlined at <http://www.php.net/manual/en/function.sprintf.php>).

That about does it for this brief look at content syndication in PHP using the XML\_RSS package on <http://pear.php.net>. With only a few short lines of code (8 to be exact, not including blank lines -- and most, if not all code was already written for you) you managed to grab a list of Flash-MX.com's most recent tutorials.

Of course this is only the tip of the PHP iceberg. In future tutorials we'll take a look at how to cache the content using some other PHP modules so your site only connects to the RSS file every hour or so (which saves on bandwidth and resources).

Happy PHP-ing!

---

Peter deHaan has been a professional web developer for the past six years and has focused heavily on integrating ColdFusion, Flash, Flash Remoting, WebServices, and XML. Over the past few years, Peter has focused more on authoring and editing books on various topics such as Flash, ColdFusion and various other web technologies while still writing articles for Flash-MX.com and helping moderate various Flash forum sites.