

# PHPGenerator

This allows you to generate XML using [PHP](#).

**This is very much a WIP, but hopefully it will be assimilated into the "real" docs – [TonyCollen](#)**

---

Part of the problem is that the PHP docs concerning compiling `--with-servlet` are not very clear. Hopefully this document will clear up some of the confusion.

This process was done on a Debian Linux 3.0r0 system and a CVS checkout of Cocoon-2.1dev

---

From the `$php_src/sapi/servlet/README:`

## Introduction

The Java Servlet SAPI Module allows for the execution of PHP 4 as a Servlet from within a Java Servlet Engine, such as Apache's Jakarta Tomcat. It builds upon the mechanism defined by PHP's Java Extension, which can be found in `/php4/ext/java`.

The primary advantage of this from a PHP perspective is that web servers which support servlets typically take great care in pooling and reusing Java Virtual Machines (JVMs).

PHP may also be bridged with the Apache Cocoon XML Publishing Framework.

A suitably configured system will invoke the PHP binaries through JNI and the output of the page will be processed through the configured XML parser and placed into the pipeline for processing by such filters as XSLT. This enables PHP developers to access the powers of Cocoon to separate their content, style, and logic without requiring them to write a single line of Java code.

While this code is intended to be able to run on any servlet engine, it has only been tested on Apache's Jakarta Tomcat to date. Bug reports, success stories and/or patches required to get this code to run on other engines would be appreciated; please send them to the PHP Development Mailinglist.

```
$ ls -l /usr/local/java
lrwxrwxrwx  1 root  staff      13 Oct 22 11:04 /usr/local/java -> j2sdk1.4.1_01

$ tar zxvf php-4.2.3.tar.gz
php-4.2.3/
php-4.2.3/build/
```

<snip/>

```
$ cd php-4.2.3
php-4.2.3$ ./configure --with-servlet --with-java=/usr/local/java
```

<snip/>

```
checking for Java support... yes
```

<this-finishes/>

\*Make sure javac is in your \$PATH, otherwise the next command will fail.

Your Java installation also needs to know about javax.servlet.\* or else it will fail as well.

You can do the following command (assuming your shell is bash) to add this package to your \$CLASSPATH if you don't have javax.servlet.\* installed system-wide:\*

```
php-4.2.3$ export CLASSPATH=$CLASSPATH:/usr/local/tomcat/common/lib/servlet.jar
```

```
php-4.2.3$ make
Making all in Zend
```

<php-compiles/>

Next, you need to put the generated phpsrvlt.jar file (It's located in \$php\_src\_dir/sapi/servlet/) into xml-cocoon2/lib/local and recompile Cocoon. When you run build, it will automatically detect the presence of the .jar file and additionally compile the PHP Generator.

After you have Cocoon deployed and running, make sure the following generator is added to your sitemap:

```
<map:generator logger="sitemap.generator.php"
  name="php"
  pool-grow="2"
  pool-max="32"
  pool-min="4"
  src="org.apache.cocoon.generation.PhpGenerator"/>
```

and then use it like so:

```
<map:match pattern="test.php">
  <map:generate type="php" src="documents/test.php"/>
  <map:transform src="stylesheets/simple-page2html.xsl"/>
  <map:serialize type="html"/>
</map:match>
```

test.php would look something like this:

```
<?php
  print "<?xml version=\"1.0\"?>\n";

  print "<page>\n";
  print "<title>Hello world from PHP</title>\n";
  print "<content>\n";
  print "<para>This is my page being dynamically generated using a PhpGenerator</para>\n";
  print "</content>\n";
  print "</page>\n";
?>
```

Hit <http://localhost:8080/cocoon/test.php> and you should have your page! Obviously, just printing XML from PHP is a little stupid, so this would be very useful if you wanted to use the myriad of other ways that PHP can generate data (Database, etc). Then again, if you're generating XML from a database, consider using [SpecificDatabaseConnection](#) that Cocoon can connect to a database.

If you already have a PHP script that generates XML, the PHPGenerator is a good way of migrating functionality to Cocoon from a LAMP-type setup.