HOW TO INSTALL PHPUNIT INTO XAMPP

by Woei Yu, May 21, 2015

Any developer worth his/her salt knows how important unit testing is. And when you are using with PHP, PHPUnit is easily the most popular Unit Testing framework.

Sometimes though PHPUnit can be hard to install on XAMPP so, in this article, I am going to cover step by step instructions on how I set up PHPUnit on my local Windows 7 machine.

Setting Up Your Environment

There are 3 steps you need to have done before installing PHPUnit.

1: Install XAMPP

First, obviously, you need to get and install *XAMPP*; if you already have it then *skip to the next step*.

XAMPP is a simple, lightweight Apache distribution that makes it extremely easy for developers to create a local web server for development and testing purposes. Everything you need to set up a Web server (Apache), database (MySQL), and programming language (PHP) – is included in a simple extractable file. XAMPP is also cross-platform, which means it works equally well on Linux, Mac and Windows.

I won't go through how to install XAMPP because it is not the focus of this article and there are *many sites to get help with installing XAMPP*.

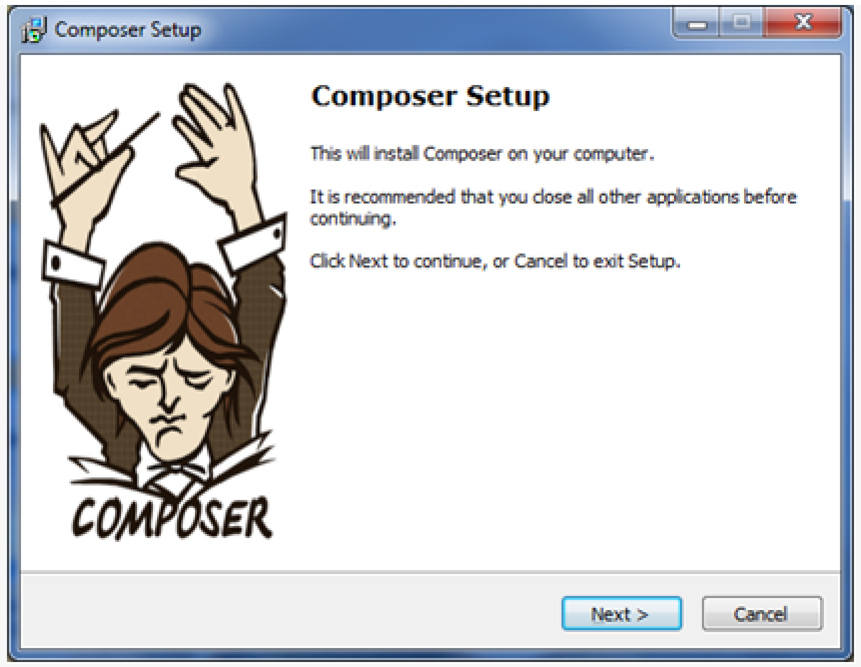
Type and Enter to Search

Woei Yu
Software Developer

HOME
MY LINKEDIN PROFILE
MY FACEBOOK PAGE
WHY I DO WHAT I DO
NEED TO REACH ME?

The next step is to download and install *Composer*; if you've already done that then *skip to the next step*.

Composer is a dependency manager that allows you to install all the things needed to run PHPUnit. Once you've downloaded the file, click on the install package.


Composer Setup

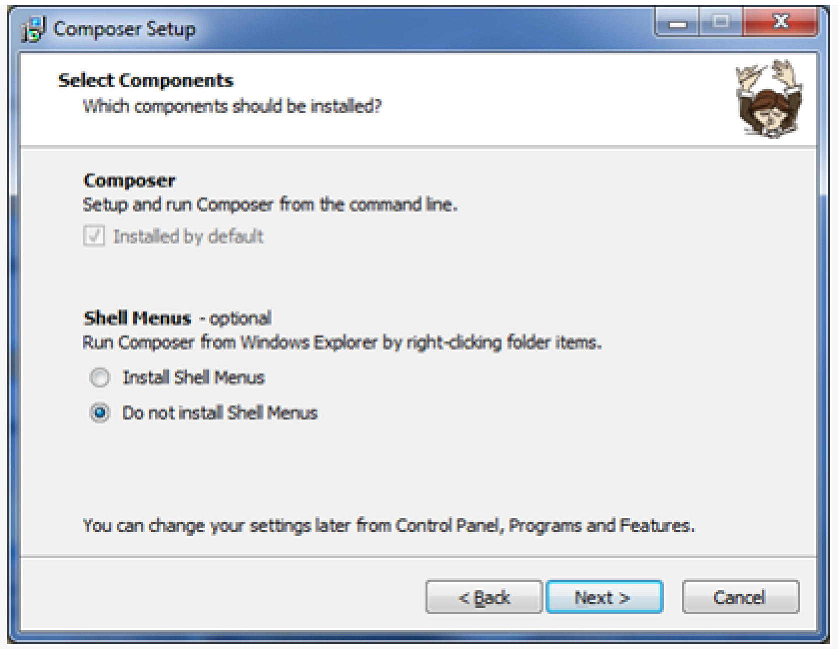
Composer Setup

This will install Composer on your computer.

It is recommended that you close all other applications before continuing.

Click Next to continue, or Cancel to exit Setup.

Step 1: Starting Setup


Composer Setup

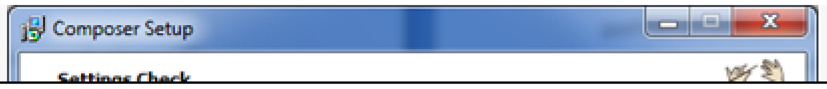
Select Components
Which components should be installed?

Composer
Setup and run Composer from the command line.
[✓] Installed by default

Shell Menus - optional
Run Composer from Windows Explorer by right-clicking folder items.
Install Shell Menus
[✓ Do not install Shell Menus]

You can change your settings later from Control Panel, Programs and Features.

Step 2: Accept the defaults


Composer Setup
Settings Check

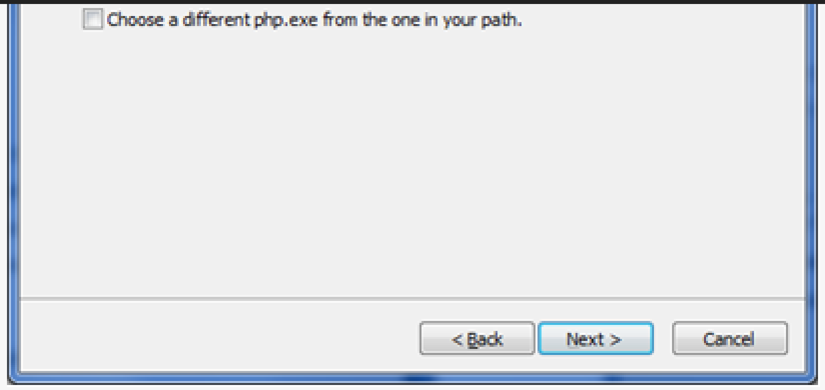
LOOKING FOR A ARTICLE?
Type and Enter to Search

RECENT POSTS
- *Fixing Mixed SSL Content Warnings on Web Pages*
- *The Dangers of Installing WordPress Plugins*
- *Free Software for Designers*
- *Layman's Guide to WordPress Security*
- *How to Install PHPUnit Into XAMPP*

RECOMMENDATIONS

LET ME KNOW WHAT YOU THINK

Type and Enter to Search



Step 3: Navigate to the php.exe file

On my machine, XAMPP is installed in c:\xampp, so my php.exe is located in **c:\xampp\php\php.exe**.

During the installation, I had this error message:

"The openssl extension is missing, which will reduce the security and stability of Composer. If possible you should enable it or recompile php with –with-openssl"

To fix that, go to php.ini file located within the PHP folder. In my particular situation, it is located in c:\xampp\php\php.ini. Search for "**extension=php_openssl.dll**".

Uncomment the line by removing the semi-colon before it, save the file, and then continue the installation.

3: Activate XDebug

XAMPP comes pre-packaged with XDebug on all versions after 1.7; to activate XDebug, we need to edit **php.ini** file.

By default, all the XDebug parameters are commented out.

```
[XDebug]
;zend_extension = "C:\xampp\php\ext\php_xdebug.dll"
;xdebug.profiler_append = 1
;xdebug.profiler_enable = 0
;xdebug.profiler_enable_trigger = 0
;xdebug.profiler_output_dir = "C:\xampp\tmp"
;xdebug.profiler_output_name = cachegrind.out.%t.%p
;xdebug.remote_enable = true
;xdebug.remote_handler = "dbgp"
;xdebug.remote_host = "localhost"
;xdebug.remote_port = 9000
;xdebug.trace_output_dir = "C:\xampp\tmp"
```

Just remove the semi-colon before all the lines to activate XDebug and save the php.ini file.

LOOKING FOR A ARTICLE?

Type and Enter to Search

RECENT POSTS

- *Fixing Mixed SSL Content Warnings on Web Pages*
- *The Dangers of Installing WordPress Plugins*
- *Free Software for Designers*
- *Layman's Guide to WordPress Security*
- *How to Install PHPUnit Into XAMPP*

RECOMMENDATIONS

LET ME KNOW WHAT YOU THINK

Type and Enter to Search

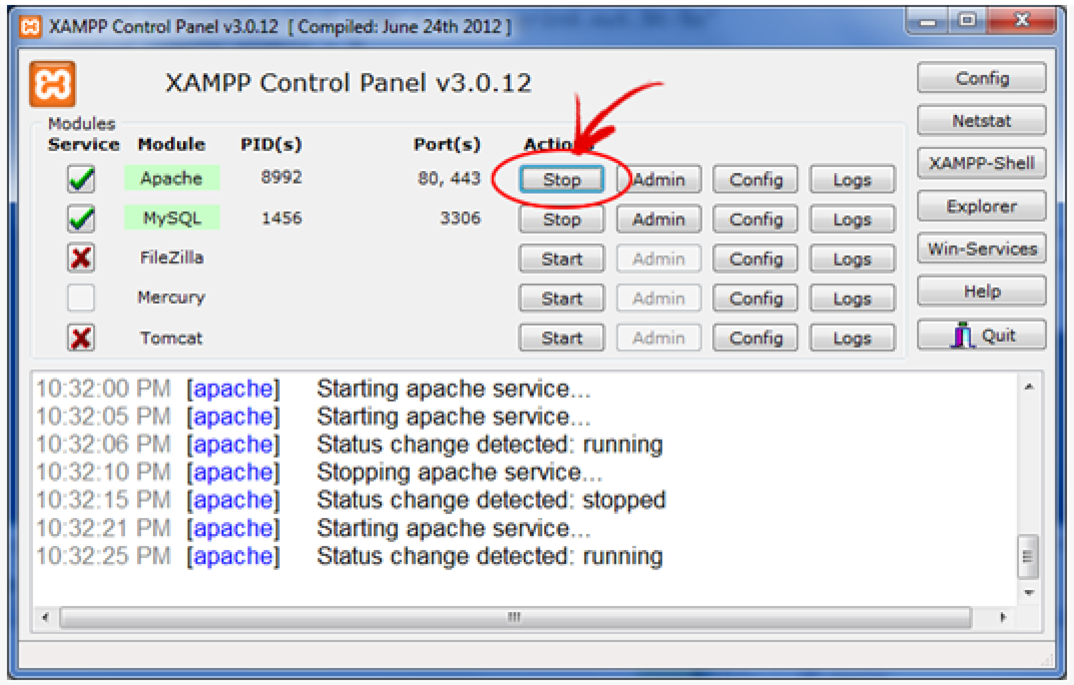
✖

Woei Yu
Software Developer

```
xdebug.profiler_output_name = cachegrind.out.%t.%p
xdebug.remote_enable = true
xdebug.remote_handler = "dbgp"
xdebug.remote_host = "localhost"
xdebug.remote_port = 9000
xdebug.trace_output_dir = "C:\xampp\tmp"
```

In order to put our new configuration to use, we'll need to restart Apache.

To do that, open your XAMPP Control Panel, and click on "**Stop**" button.

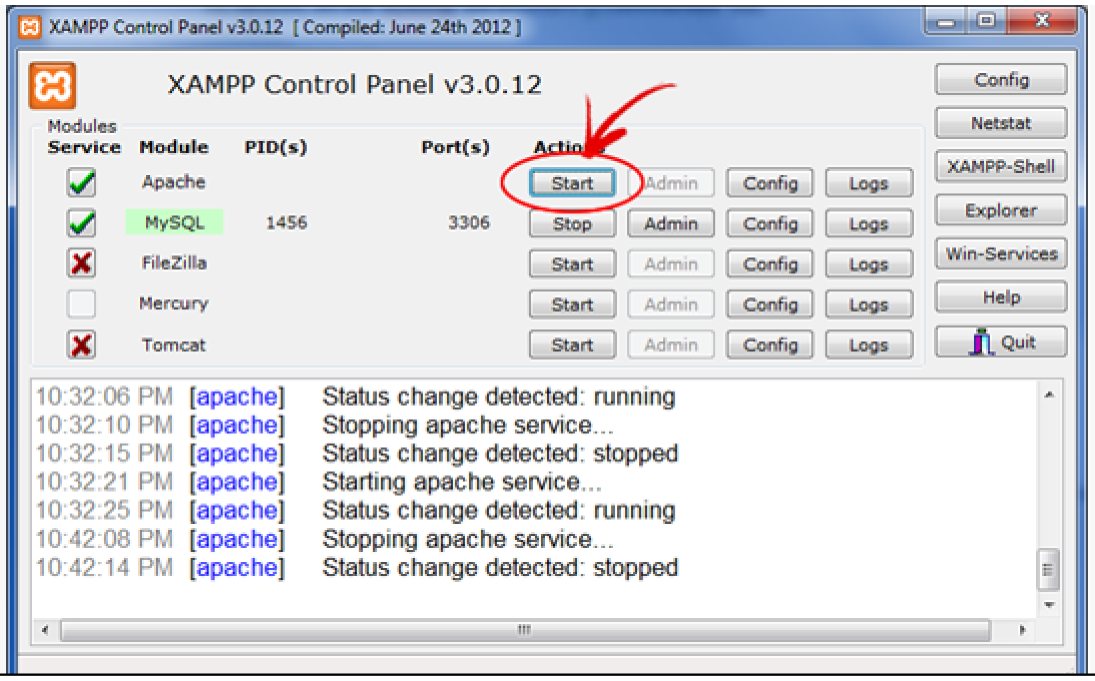

XAMPP Control Panel v3.0.12 [Compiled: June 24th 2012]
XAMPP Control Panel v3.0.12
Config
Netstat
XAMPP-Shell
Explorer
Win-Services
Help
Quit
Stop
Start
Admin
Logs

Service	Module	PID(s)	Port(s)
✓	Apache	8992	80, 443
✓	MySQL	1456	3306
✗	FileZilla		
	Mercury		
✗	Tomcat		

```
10:32:00 PM  [apache]   Starting apache service...
10:32:05 PM  [apache]   Starting apache service...
10:32:06 PM  [apache]   Status change detected: running
10:32:10 PM  [apache]   Stopping apache service...
10:32:15 PM  [apache]   Status change detected: stopped
10:32:21 PM  [apache]   Starting apache service...
10:32:25 PM  [apache]   Status change detected: running
```

To Stop the XAMPP Server

Then click on the "**Start**" button to restart the server.


XAMPP Control Panel v3.0.12 [Compiled: June 24th 2012]
XAMPP Control Panel v3.0.12
Config
Netstat
XAMPP-Shell
Explorer
Win-Services
Help
Quit
Start
Stop
Admin
Logs

Service	Module	PID(s)	Port(s)
✓	Apache		
✓	MySQL	1456	3306
✗	FileZilla		
	Mercury		
✗	Tomcat		

```
10:32:06 PM  [apache]   Status change detected: running
10:32:10 PM  [apache]   Stopping apache service...
10:32:15 PM  [apache]   Status change detected: stopped
10:32:21 PM  [apache]   Starting apache service...
10:32:25 PM  [apache]   Status change detected: running
10:42:08 PM  [apache]   Stopping apache service...
10:42:14 PM  [apache]   Status change detected: stopped
```

LOOKING FOR A ARTICLE?

Type and Enter to Search

RECENT POSTS

- *Fixing Mixed SSL Content Warnings on Web Pages*
- *The Dangers of Installing WordPress Plugins*
- *Free Software for Designers*
- *Layman's Guide to WordPress Security*
- *How to Install PHPUnit Into XAMPP*

RECOMMENDATIONS

LET ME KNOW WHAT YOU THINK

Type and Enter to Search

Woei Yu
Software Developer

Installing PHPUnit

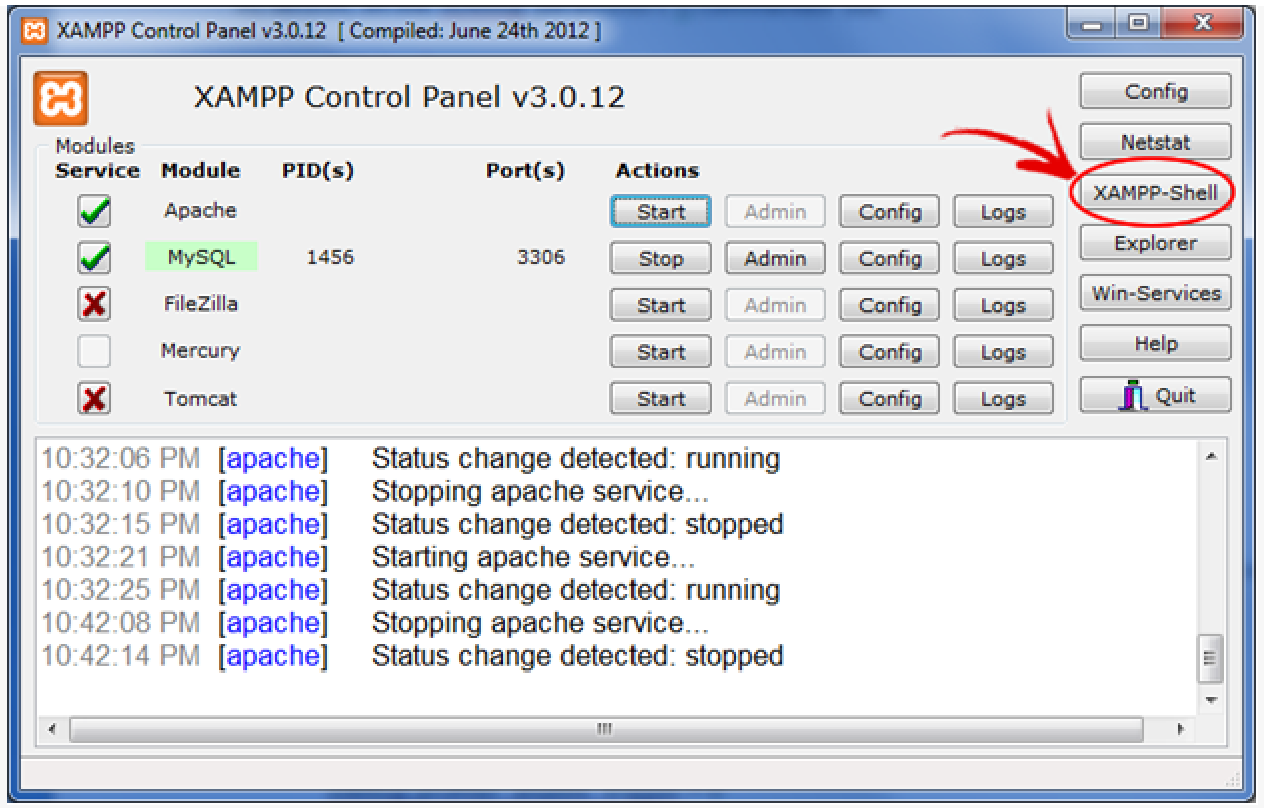
Now that everything is setup it's time to install PHPUnit.

All web files typically sit inside the "htdocs" directory located within the XAMPP directory. So, on my machine, my web files should be placed in **c:\xampp\htdocs**.

To install PHPUnit using Composer we need the file composer.json which you can *download here*. Open the file and, you see in line 3, I defined phpunit to be required. Once the install is run, composer will download the requested files and place them in the "**vendor**" directory.

```
1 {
2      "require"     : { },
3      "require-dev" : { "phpunit/phpunit": "*" }
4 }
```

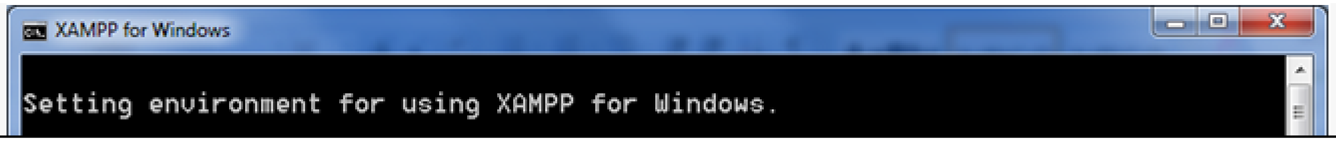
To install PHPUnit, we fire up the Shell program from XAMPP Control Panel.



```
10:32:06 PM  [apache]   Status change detected: running
10:32:10 PM  [apache]   Stopping apache service...
10:32:15 PM  [apache]   Status change detected: stopped
10:32:21 PM  [apache]   Starting apache service...
10:32:25 PM  [apache]   Status change detected: running
10:42:08 PM  [apache]   Stopping apache service...
10:42:14 PM  [apache]   Status change detected: stopped
```

Open the Shell Program in XAMPP

I have placed the **composer.json** file within the "**UnitTest**" folder and I need to navigate to that directory where my tests will be by typing in "**chdir htdocs\UnitTest**".



```
Setting environment for using XAMPP for Windows.
```

LOOKING FOR A ARTICLE?

Type and Enter to Search

RECENT POSTS

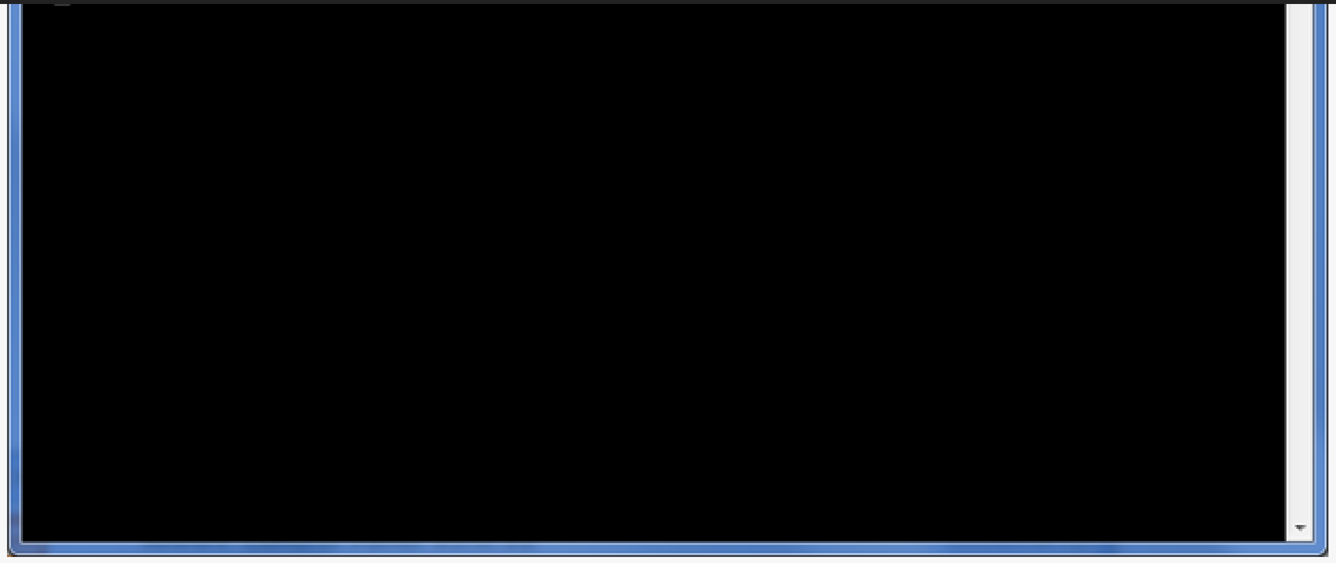
- *Fixing Mixed SSL Content Warnings on Web Pages*
- *The Dangers of Installing WordPress Plugins*
- *Free Software for Designers*
- *Layman's Guide to WordPress Security*
- *How to Install PHPUnit Into XAMPP*

RECOMMENDATIONS

LET ME KNOW WHAT YOU THINK

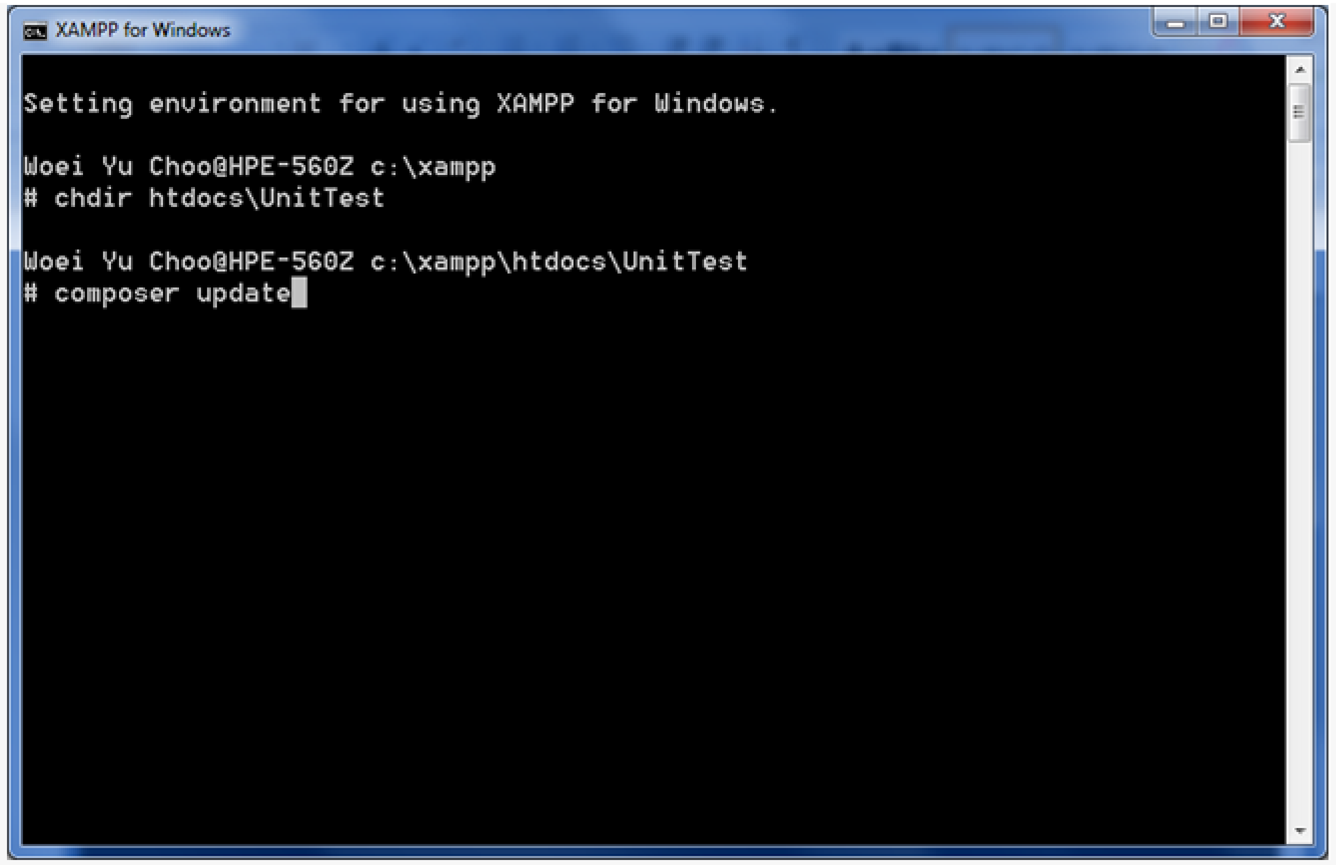
Type and Enter to Search

Woei Yu
Software Developer



Navigate to Directory in Shell Program

Now I'm in the directory of UnitTest folder, I can run "**composer update**"… and my Composer repository should update by reading the **composer.json** file and installing everything it needs to run PHPUnit and putting the necessary dependency files within the "**vendor**" directory.



```
Setting environment for using XAMPP for Windows.

Woei Yu Choo@HPE-560Z c:\xampp
# chdir htdocs\UnitTest

Woei Yu Choo@HPE-560Z c:\xampp\htdocs\UnitTest
# composer update
```

Install PHPUnit in Shell Program

LOOKING FOR A ARTICLE?

Type and Enter to Search

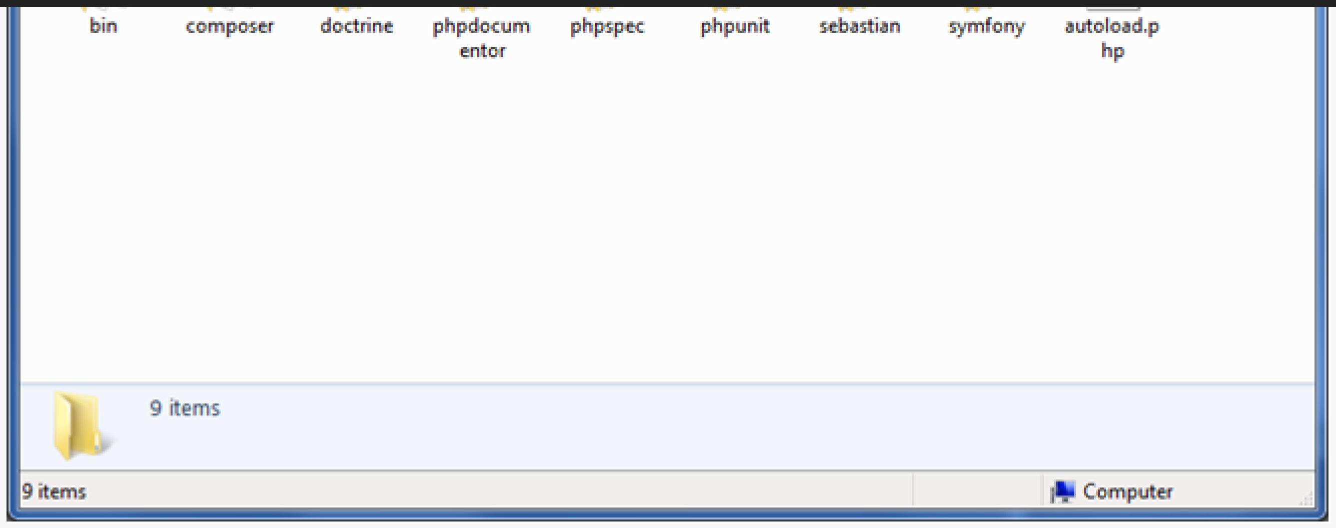
RECENT POSTS

- *Fixing Mixed SSL Content Warnings on Web Pages*
- *The Dangers of Installing WordPress Plugins*
- *Free Software for Designers*
- *Layman's Guide to WordPress Security*
- *How to Install PHPUnit Into XAMPP*

RECOMMENDATIONS

LET ME KNOW WHAT YOU THINK

Type and Enter to Search



Installed PHPUnit Files and Folders

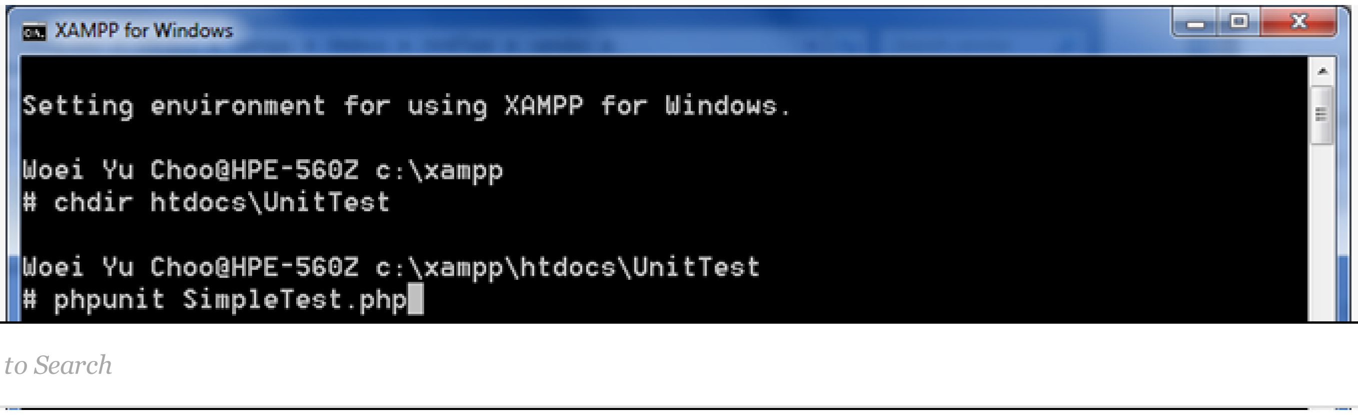
Now that PHP Unit is installed we can start using it.

Using PHPUnit

I have created a php file, SimpleTest.php, that has 2 assertions. Assertions are the basis of unit testing in PHP and are basically methods that verify that a unit of code behaves as expected. In this case, I am returning a false and a true assertion.

```php
<?php

class SimpleTest extends \PHPUnit_Framework_TestCase
{
        public function testSimple1()
        {
         $this->assertEquals(2, 1 + 2);
        }
        public function testSimple2()
        {
         $this->assertEquals(2, 1+1);
        }
}
```

So to test the file SimpleTest.php, just type in phpunit SimpleTest.php; note that PHPUnit will assume that all public methods in the class are tests and will automatically execute them.



```
Setting environment for using XAMPP for Windows.

Woei Yu Choo@HPE-560Z c:\xampp
# chdir htdocs\UnitTest

Woei Yu Choo@HPE-560Z c:\xampp\htdocs\UnitTest
# phpunit SimpleTest.php
```

LOOKING FOR A ARTICLE?

Type and Enter to Search

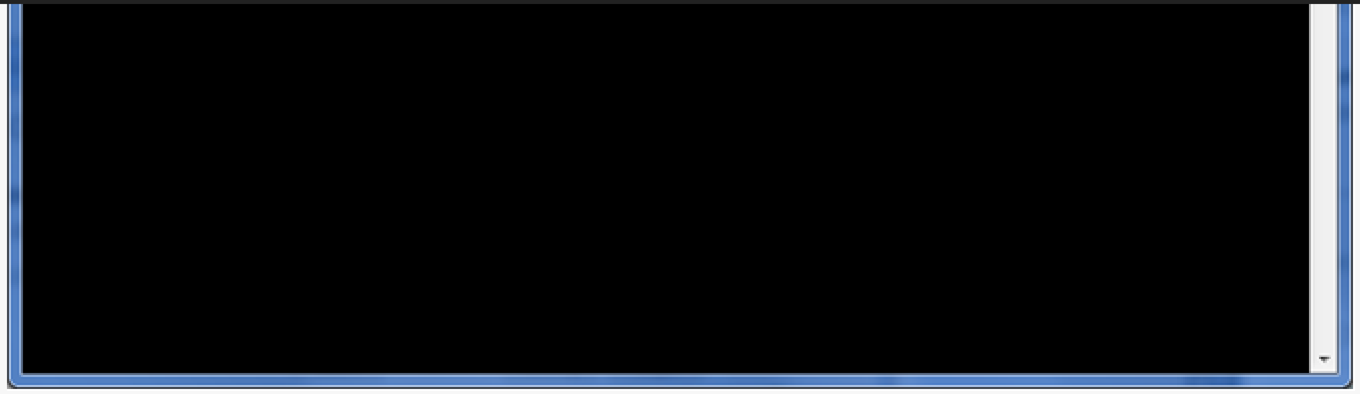
RECENT POSTS

- *Fixing Mixed SSL Content Warnings on Web Pages*
- *The Dangers of Installing WordPress Plugins*
- *Free Software for Designers*
- *Layman's Guide to WordPress Security*
- *How to Install PHPUnit Into XAMPP*

RECOMMENDATIONS

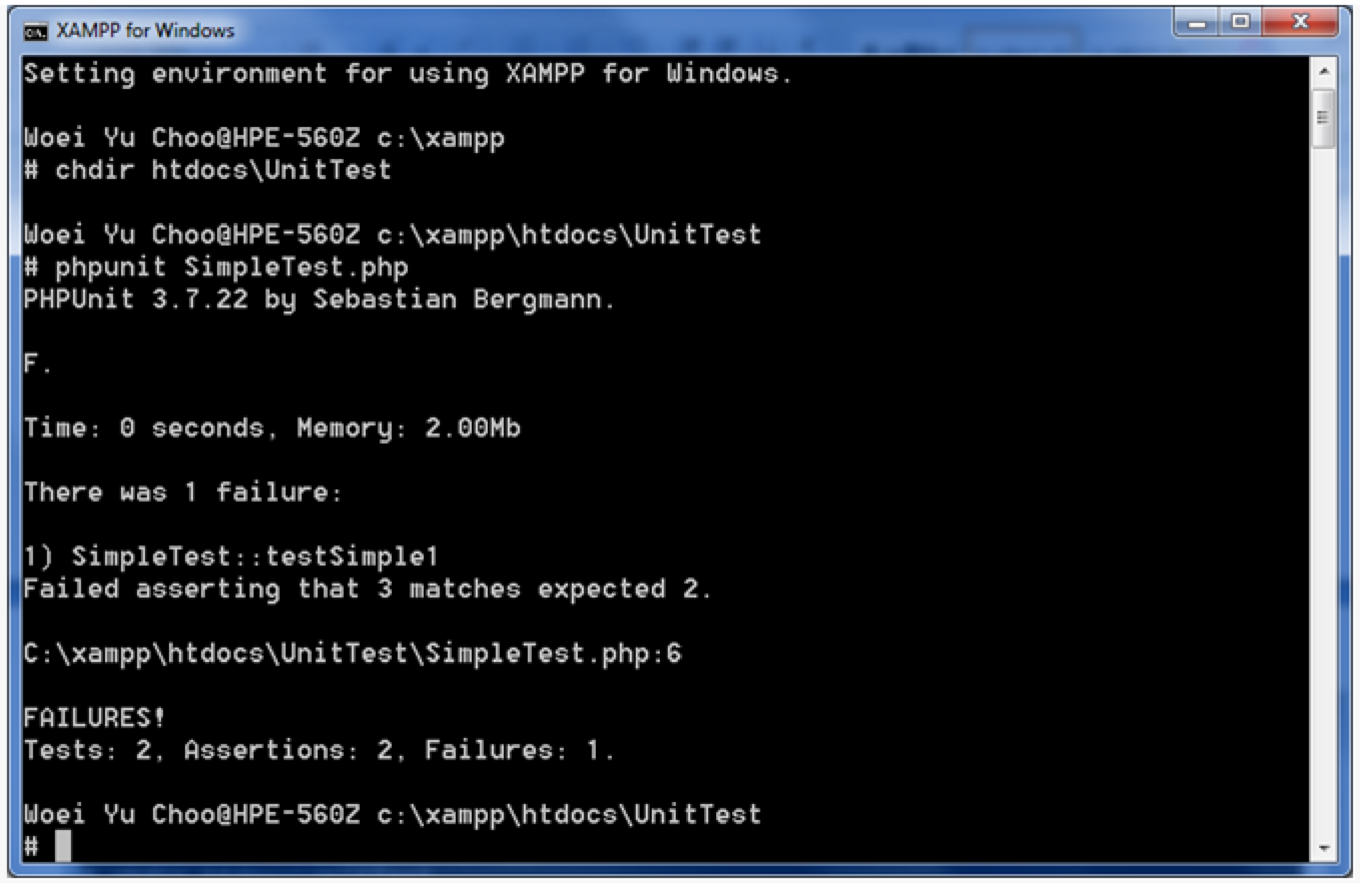
LET ME KNOW WHAT YOU THINK

Type and Enter to Search



Simple PHPUnit Test

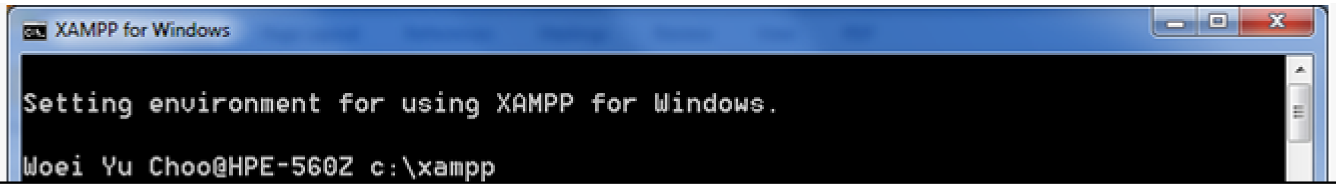
And it returns the right test results with 2 assertions and 1 failure.



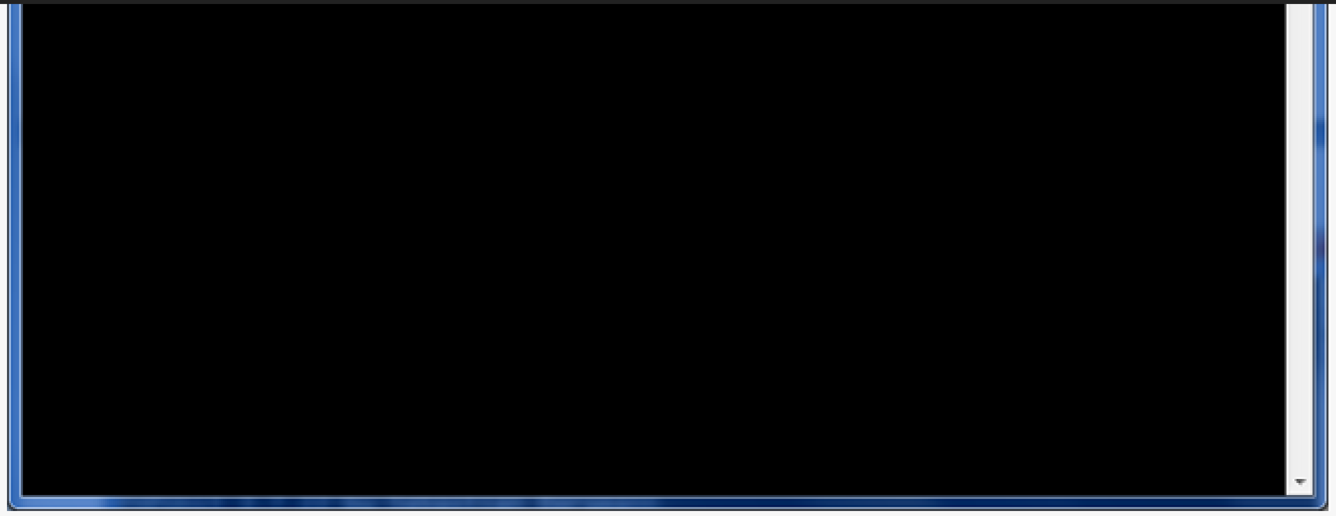
```
Setting environment for using XAMPP for Windows.

Woei Yu Choo@HPE-560Z c:\xampp
# chdir htdocs\UnitTest

Woei Yu Choo@HPE-560Z c:\xampp\htdocs\UnitTest
# phpunit SimpleTest.php
PHPUnit 3.7.22 by Sebastian Bergmann.

F.

Time: 0 seconds, Memory: 2.00Mb

There was 1 failure:

1) SimpleTest::testSimple1
Failed asserting that 3 matches expected 2.

C:\xampp\htdocs\UnitTest\SimpleTest.php:6

FAILURES!
Tests: 2, Assertions: 2, Failures: 1.

Woei Yu Choo@HPE-560Z c:\xampp\htdocs\UnitTest
#
```

Test Result for a Simple PHPUnit Test

And if you want to do a unit test outside of your test folder, simple type in phpunit followed by relative path of your test file. In this case, I created a simple test named "OutsideSimpleTest" which I placed under "htdocs" folder. So the command will be "phpunit ../OutsideSimpleTest.php"



```
Setting environment for using XAMPP for Windows.

Woei Yu Choo@HPE-560Z c:\xampp
```


A Simple PHPUnit Test to a Specific File Path

So you can see you are not constrained within the folder where you installed phpUnit.

But this can be rather tedious not to mention inefficient to do test one page at a time, which is why it is so the next section is a huge time saver. Running through all your tests in a batch.

Running Batch Files

First, we need to define our test suite configuration in the phpunit.xml file and place it in my PHPUnit folder. In this case, we specify that our "SimpleProject" folder will contain all the test files. Here is how to do that:

```
<?xml version="1.0" encoding="UTF-8"?>
<phpunit colors="true" bootstrap="vendor/autoload.php">
    <testsuites>
        <testsuite name="WY's Test Suite">
            <directory>./SimpleProject/</directory>
        </testsuite>
    </testsuites>
</phpunit>
```

<directory>./SimpleProject/</directory> simple tells PHPUnit where your test will be located so you don't have to manually tell it every time you run your test.

By default, anything in that folder that has the suffix "test" in the file name, will be run as a test. If you have a different naming convention, eg you want to name your test like Simple1.UnitTest.php, use this

```
<directory suffix=".UnitTest.php">Custom</directory>
```

To verify that, I have created 2 test files (Simple1Test.php and Simple2Test.php), both returning true assertions

LOOKING FOR A ARTICLE?

Type and Enter to Search

RECENT POSTS

- *Fixing Mixed SSL Content Warnings on Web Pages*
- *The Dangers of Installing WordPress Plugins*
- *Free Software for Designers*
- *Layman's Guide to WordPress Security*
- *How to Install PHPUnit Into XAMPP*

RECOMMENDATIONS

LET ME KNOW WHAT YOU THINK

Type and Enter to Search

Woei Yu
Software Developer

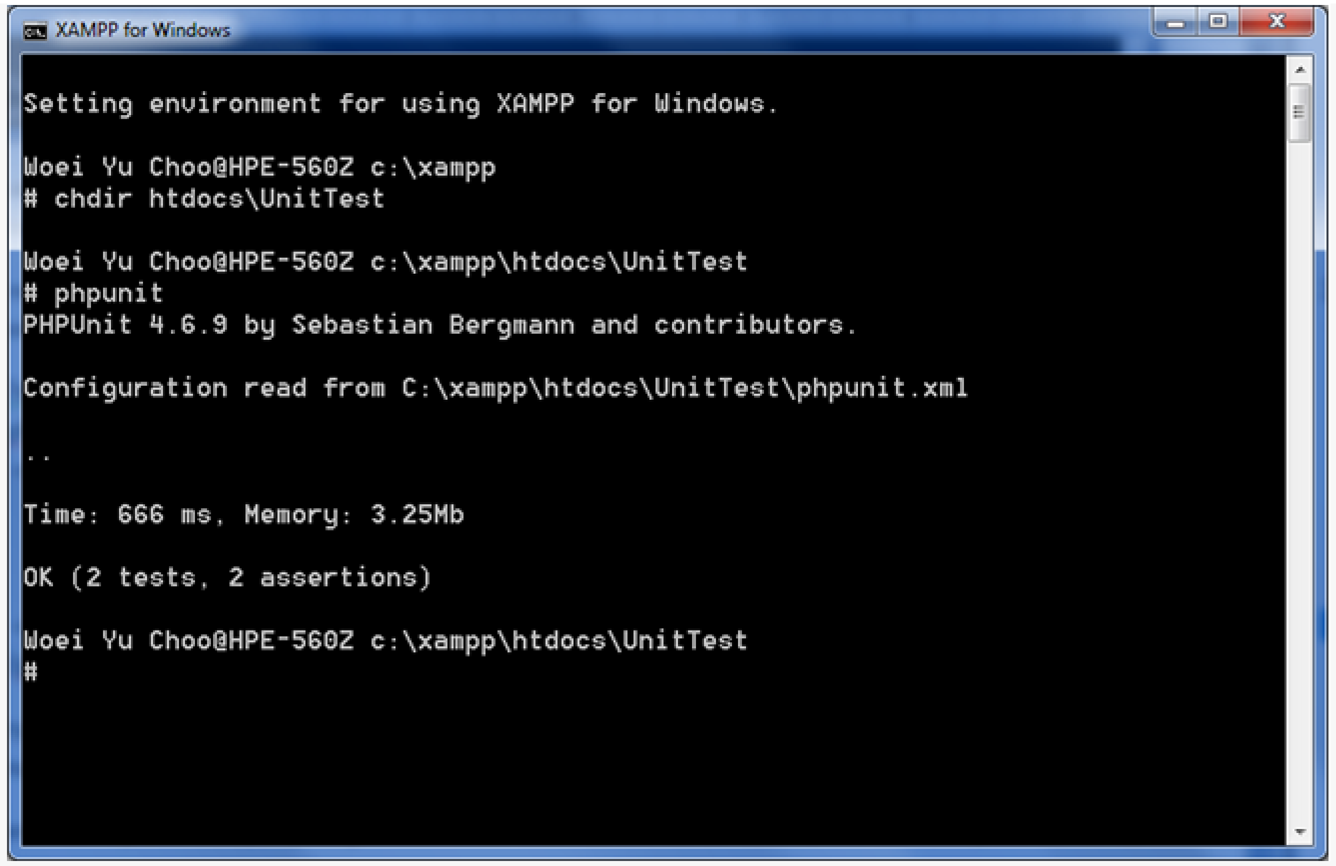
HOME
MY LINKEDIN PROFILE
MY FACEBOOK PAGE
WHY I DO WHAT I DO
NEED TO REACH ME?

```php
    public function testSimple()
    {
        $this->assertEquals(2, 1 + 1);
    }

}
```

And ...

```php
<?php
class Simple2Test extends \PHPUnit_Framework_TestCase
{
    public function testSimple()
    {
        $this->assertEquals(2, 3 - 1);
    }

}
```

Simple type in "**phpunit**". It will read in the phpunit.xml and load in the appropriate test cases and run them.



```
Setting environment for using XAMPP for Windows.

Woei Yu Choo@HPE-560Z c:\xampp
# chdir htdocs\UnitTest

Woei Yu Choo@HPE-560Z c:\xampp\htdocs\UnitTest
# phpunit
PHPUnit 4.6.9 by Sebastian Bergmann and contributors.

Configuration read from C:\xampp\htdocs\UnitTest\phpunit.xml

..

Time: 666 ms, Memory: 3.25Mb

OK (2 tests, 2 assertions)

Woei Yu Choo@HPE-560Z c:\xampp\htdocs\UnitTest
#
```

Runing PHPUnit via phpunit.xml configuration file

The best thing about configuration files you have a range of settings you can define. If you want to test certain files instead of running all the files under a particular directory ...

LOOKING FOR A ARTICLE?

Type and Enter to Search

RECENT POSTS

- *Fixing Mixed SSL Content Warnings on Web Pages*
- *The Dangers of Installing WordPress Plugins*
- *Free Software for Designers*
- *Layman's Guide to WordPress Security*
- *How to Install PHPUnit Into XAMPP*

RECOMMENDATIONS

LET ME KNOW WHAT YOU THINK

Type and Enter to Search

```
    </testsuites>
</phpunit>
```

If you want to include or exclude files, you can invoke the blacklist and whitelist settings

```
<filter>
  <blacklist>
    <directory suffix=".php">/path/to/files</directory>
    <file>/path/to/file</file>
    <exclude>
      <directory suffix=".php">/path/to/files</directory>
      <file>/path/to/file</file>
    </exclude>
  </blacklist>
  <whitelist processUncoveredFilesFromWhitelist="true">
    <directory suffix=".php">/path/to/files</directory>
    <file>/path/to/file</file>
    <exclude>
      <directory suffix=".php">/path/to/files</directory>
      <file>/path/to/file</file>
    </exclude>
  </whitelist>
</filter>
```

The available settings are pretty noteworthy. See *https://phpunit.de/manual/current/en/appendixes.configuration.html* for more XML information.

Conclusion

To recap, to install PHPUnit into XAMPP…

1. Make sure XAMPP is installed
2. Make sure Composer is installed
3. Make sure XDebug is activated
4. Create the composer.json and install PHPUnit
5. Create the PHPUnit.xml configuration file and the simple test case
6. Run the PHPUnit on the simple test case to make sure PHPUnit is working

Download source file here

6 COMMENTS

MICHAEL WEST ▪ REPLAY
November 22, 2015

LOOKING FOR A ARTICLE?
Type and Enter to Search

RECENT POSTS
- *Fixing Mixed SSL Content Warnings on Web Pages*
- *The Dangers of Installing WordPress Plugins*
- *Free Software for Designers*
- *Layman's Guide to WordPress Security*
- *How to Install PHPUnit Into XAMPP*

RECOMMENDATIONS

LET ME KNOW WHAT YOU THINK

Type and Enter to Search

Installing PHPunit on Windows

Installing PHPunit basically requires downloading one single `.phar` file, which can be executed with PHP. You can find it on the PHPunit Getting Started section of their page, or simply save the following link: https://phar.phpunit.de/phpunit.phar.

Writing PHPunit Tests

There is an elaborate example in the PHPunit Getting Started section with OOP code, but I will quickly go through a very simple one below

```
.:
my_class.php  my_functions.php  phpunit.phar  README.md  tests

.\tests:
my_class_test.php  my_function_test.php
```

- `my_functions.php` is the procedural part of code we want to test
- `my_class.php` is the OOP part of code we want to test
- `phpunit.phar` is the PHPunit test suite
- `tests` is the directory that contains our tests
- `tests/my_function_test.php` is the file that contains the tests we write
- `tests/my_function_test.php` is the file that contains the tests we write

The content of my `my_functions.php` file is the following:

```
 <?php
    function my_addition($arg1, $arg2){
        return $arg1 + $arg2;
    }
?>
```

To thest this function, I can simply use the following code in `tests/my_function_test.php`:

```
 <?php

class MyProceduralTest extends PHPUnit_Framework_Testcase {
```

```
class MyProceduralTest extends PHPunit_Framework_TestCase {

    /*
     * Testing the addition function
     */

    public function testAddition(){
        include('my_functions.php'); // must include if tests are for non
OOP code
        $result = my_addition(1,1);
        $this->assertEquals(2, $result);
    }
}
?>
```

Running the PHPunit Tests on Windows

To run the unit tests, you need to open the command line interface on Windows. Most easily you can do that by opening your start menu and typing `cmd` in the search box.

Firstly we will need to switch to the directory where the code, that is supposed to be tested and the tests themselves are placed.

If you're working with WAMP, your path could be something like this: `C:\wamp\www\yourproject`. To get to that path we simply type in the cmd: `cd C:\wamp\www\yourproject`.

Now we can start running our tests:

```
C:\php5\php.exe phpunit.phar tests\my_function_test.php
```

If you want to test a class, you don't need to do any `include()` in your test file and can run a command like the following:

```
C:\php5\php.exe phpunit.phar --bootstrap my_class.php
tests\my_class_test.php
```

This for example is the output of running the `my_function_test.php`:

```
phpunit-demo (master) $ php phpunit.phar tests\my_function_test.php
PHPUnit 3.7.32 by Sebastian Bergmann.

.

Time: 83 ms, Memory: 2.50Mb

OK (1 test, 1 assertion)
```

OK! Our first test passed! Easy, right? Right! Now if we change the value that we expect to get if we add 1 and 1, we will receive an error message, that tells us which test fails. This is extremely handy if you're not testing one, but 500 functions.

Changing:

```
         $result = my_addition(1,1);
-        $this->assertEquals(3, $result);
+        $this->assertEquals(3, $result);
```

Error Message:

```
PHPUnit 3.7.32 by Sebastian Bergmann.

F

Time: 272 ms, Memory: 2.75Mb

There was 1 failure:

1) MyProceduralTest::testAddition
Failed asserting that 2 matches expected 3.

/home/geronimo/public_html/phpunit-demo/tests/my_function_test.php:12

FAILURES!
Tests: 1, Assertions: 1, Failures: 1.
```

The most valuable line is probably `Failed asserting that 2 matches expected 3.`, because it tells both the return value and which one we thought we would get in the test.

Writing Your Own PHP Tests

To write your own tests, you can have a look at the list of all assertions for PHPUnit. It has a lot of practial functions that can test for `true` or `false`, if a function outputs a number or if a value is lesser or greater than another. All assertion functions come with handy examples, from which you can build your own tests.

The functions may seem very basic, but since it also supports regulary expressions, you can test for many of the string conversions that are typical for web projects, or url escaping, user input validation and so on.