



Frank Kleine, 1&1 Internet AG

vfsStream

A better approach for file system dependent tests





1&1





vfs://Stream



vfs://Stream







T-Shirt available at zazzle.de
(no, I'm not paid for this)





(Obligatory Zen-Style image)



AND NOW FOR
SOMETHING
COMPLETELY DIFFERENT

Unit tests

- I assume you do unit tests
- Testing file system related functions/methods can be hard
- `setup()`, `teardown()`, and unusual scenarios

Basic example: a method to test

```
class Example {  
    public function __construct($id) {  
        $this->id = $id;  
    }  
  
    public function setDirectory($dir) {  
        $this->dir = $dir . '/' . $this->id;  
        if (file_exists($this->dir) === false) {  
            mkdir($this->dir, 0700, true);  
        }  
    }  
    ...  
}
```

Basic example: traditional test

```
$DIR = dirname(__FILE__);  
public function setUp() {  
    if (file_exists($DIR . '/id')) {  
        rmdir($DIR . '/id');  
    }  
}  
public function tearDown() {  
    if (file_exists($DIR . '/id')) {  
        rmdir($DIR . '/id');  
    }  
}  
public function testDirectoryIsCreated() {  
    $example = new Example('id');  
    $this->assertFalse(file_exists($DIR . '/id'));  
    $example->setDirectory($DIR);  
    $this->assertTrue(file_exists($DIR . '/id'));  
}
```

Basic example: vfsStream test

```
public function setUp() {
    vfsStreamWrapper::register();
    $root = new vfsStreamDirectory('aDir');
    vfsStreamWrapper::setRoot($root);
}

public function testDirectoryIsCreated() {
    $url = vfsStream::url('aDir/id');
    $example = new Example('id');
    $this->assertFalse(file_exists($url));
    $example->setDirectory(vfsStream::url('aDir'));
    $this->assertTrue(file_exists($url));
}
```


Advantages

- Cleaner tests
- Nothing happens on disc, all operations are memory only
- More control over file system environment

How does this work?

- PHP stream wrapper
 - ⦿ Allows you to invent your own url protocol
- Whenever a `vfs://` url is given to a file system function PHP calls back the registered stream implementation for this protocol
- Works well for most of file system functions

What does not work?

- File system functions working with pure file names only:
 - ⦿ `realpath()`
 - ⦿ `touch()`
- File system function without stream wrapper support:
 - ⦿ `chmod()`
 - ⦿ `chown()`
 - ⦿ `chgrp()`
- `ext/zip` does not support userland stream wrappers

Example with file mode

```
class Example {
    public function __construct($id, $mode = 0700) {
        $this->id    = $id;
        $this->mode  = $mode;
    }

    public function setDirectory($dir) {
        $this->dir = $dir . '/' . $this->id;
        if (file_exists($this->dir) === false) {
            mkdir($this->directory, $this->mode, true);
        }
    }
    ...
}
```

Example with file mode, cont.

```
$DIR = dirname(__FILE__);
public function testDirDefaultFilePermissions() {
    $example = new Example('id');
    $example->setDirectory($DIR);
    if (DIRECTORY_SEPARATOR === '\\') {
        $this->assertEquals(40777, decoct(fileperms($DIR . '/id')));
    } else {
        $this->assertEquals(40700, decoct(fileperms($DIR . '/id')));
    }
}

public function testDirDifferentFilePermissions() {
    $example = new Example('id', 0755);
    $example->setDirectory($DIR);
    if (DIRECTORY_SEPARATOR === '\\') {
        $this->assertEquals(40777, decoct(fileperms($DIR . '/id')));
    } else {
        $this->assertEquals(40755, decoct(fileperms($DIR . '/id')));
    }
}
```

Example with file mode, cont.

```
public function setUp() {
    vfsStreamWrapper::register();
    $this->root = new vfsStreamDirectory('aDir');
    vfsStreamWrapper::setRoot($this->root);
}

public function testDirDefaultFilePermissions() {
    $example = new Example('id');
    $example->setDirectory(vfsStream::url('aDir'));
    $this->assertEquals(0700,
        $this->root->getChild('id')->getPermissions());
}

public function testDirDifferentFilePermissions() {
    $example = new Example('id', 0755);
    $example->setDirectory(vfsStream::url('aDir'));
    $this->assertEquals(0755,
        $this->root->getChild('id')->getPermissions());
}
```

Advantages

- Independent of operating system a test is running on
- Intentions of test cases become more clear
- Easier to understand

Different config files

```
class RssFeedController {
    public function __construct($configPath) {
        $feeds = Properties::fromFile($configPath . '/rss-feeds.ini')
            ->getSection('feeds', array());
        if (count($feeds) === 0) {
            throw new ConfigurationException();
        }

        $this->routeName = valueFromRequest();
        if (null === $this->routeName) {
            // no special feed requested, use first configured one
            reset($feeds);
            $this->routeName = key($feeds);
        }
    }
    ...
}
```


Different config files, cont.

```
public function setUp() {
    vfsStreamWrapper::register();
    $root = new vfsStreamDirectory('config');
    vfsStreamWrapper::setRoot($root);
    $this->configFile = vfsStream::newFile('rss-feeds.ini')
                                                ->at($root);
}

/**
 * @test
 * @expectedException FileNotFoundException
 **/
public function loadFeedsFailsIfFeedConfigFileDoesNotExist() {
    $example = new RssFeedController(vfsStream::url('doesNotExist'));
}
```

Different config files, cont. 2

```
/**
 * @test
 * @expectedException ConfigurationException
 **/
public function noFeedsSectionConfiguredThrowsException() {
    $this->configFile->setContent('');

    $example = new RssFeedController(vfsStream::url('config'));
}
```

Different config files, cont. 3

```
/**
 * @test
 * @expectedException ConfigurationException
 **/
public function noFeedsConfiguredThrowsException() {
    $this->configFile->setContent('[feeds]');

    $example = new RssFeedController(vfsStream::url('config'));
}
```

Different config files, cont. 4

```
/**
 * @test
 **/
public function selectsFirstFeedIfNoneGivenWithRequestValue() {
    $this->configFile->setContent('[feeds]\n
default = "org::stubbles::test::xml::rss::DefaultFeed"');

    $example = new RssFeedController(vfsStream::url('config'));

    // assertions that the default feed was selected

    ...
}
```

Different config files, cont. 5

```
/**
 * @test
 **/
public function selectsOtherFeedBasedOnRequestValue() {
    $this->configFile->setContent("[feeds]\n
default = \"org::stubbles::test::xml::rss::DefaultFeed\"\n
other = \"org::stubbles::test::xml::rss::OtherFeed\"\n");

    $example = new RssFeedController(vfsStream::url('config'));

    // assertions that the other feed was selected

    ...
}
```

Advantages

- Concentrate on the test, not on config files
- Everything is in the test, no test related information in external files
- Simple to set up
- Create different test scenarios on the fly

vfsStream 0.7.0

- Considers file permissions correctly
- Allows to write tests to check how your application responds to incorrect file permissions
- To be released in the next days
- Probably buggy :->

File permissions

```
class Example {  
    public function writeConfig($config, $configFile) {  
        file_put_contents($configFile, serialize($config));  
    }  
  
    ...  
}
```


File permissions, the tests

```
/**
 * @test
 */
public function normalTest() {
    vfsStreamWrapper::setRoot(vfsStream::newDirectory('exampleDir'));
    $example = new FilePermissionsExample();
    $example->writeConfig(array('foo' => 'bar'),
                          vfsStream::url('exampleDir/writable.ini')
    );

    // assertions here
}
```

File permissions, another test

```
/**
 * @test
 */
public function directoryNotWritable() {
    vfsStreamWrapper::setRoot(
        vfsStream::newDirectory('exampleDir', 0444)
    );
    $example = new FilePermissionsExample();
    $example->writeConfig(array('foo' => 'bar'),
        vfsStream::url('exampleDir/config.ini')
    );
}
```

Advantages

- Allows simple tests you would not do otherwise because they are too hard to do
- Helps to find probably buggy code or code which behaves not user friendly
- Makes your application more failsafe



Ressources

- <http://vfs.bovigo.org/>
- <http://php.net/class.streamwrapper>
- Twitter
 - @bovigo
 - @mikey179



