

Monday, December 3, 2007

PHPMutagen: Mutation Testing for PHP5

I swear - this is the last bdd/testing related idea I'm having for a long time 

. There's (so far) PHPSpec, PHPMock and now PHPMutagen. The first is a Behaviour-Driven Development framework, the other two are unrelated libraries intended for use with PHPSpec, SimpleTest, PHPUnit or any other conceivable testing framework.

As I said in my previous entry I was thinking about how to write a Mutation Testing engine. The "braindump" was to use PHP's built in Tokenizer to break down a class file into digestable pieces which could be mutated, and then reconstructed into a mutated file. Once you allow for a working copy of the original source (let's not mutate the original!) it turned out to be a reasonable approach.

The source code is still rough, but essentially operational. I wouldn't get too excited until I find some time to scan through it again (after I get over this cold) and get the spec coverage into shape. For the moment I'll tease you all with some sample output.

Here's what you really want to see after running PHPMutagen (note this was only possible with PHPSpec, but a PHPUnit or PHPT adapter would be equally simple to add later):

```
.....  
7 Mutants born out of the mutagenic slime!
```

```
7 Mutants exterminated!
```

```
No Mutants survived! Muahahahaha!
```

Here's what you probably want not to see 

. The comment at the end is important - in Mutation Testing there will always be a few elusive ghosts which aren't a real problem because you can switch operators in code without altering behaviour in some circumstances. Using `- $this->_actualCallCount++;`
`+ $this->_actualCallCount--;`
`if (!is_null($this->_exceptionToThrow)) {`
`if (is_array($this->_exceptionToThrow)) {`
`$class = $this->_exceptionToThrow[0];`
`$message = $this->_exceptionToThrow[1];`

Happy Hunting! Remember that some Mutants may just be Ghosts (or if you want to be boring, false positives).

The diff output for a mutant is courtesy of PEAR::Text_Diff's unified diff renderer and a little extra decorating.

Posted by Pádraic Brady in PHP General, PHP Security at 13:38


Hi Padraic,

You might want to consider using a variant of http://pear.php.net/PHP_Parser to do your mutation, this way you can more reliably determine where you can mutate and even offer more advanced structural changes. Anonymous on Dec 3 2007, 18:27

Thanks for the link Greg and I'll check it out. The code so far is a big Tracer to get some of the theory into code quickly but I'll be ripping out the dodgy segments and replacing them with a spec'd out robust version over the next week or so. Anonymous on Dec 3

2007, 21:42

like "Heckle":<http://ruby.sadi.st/Heckle.html> Anonymous on Dec 5 2007, 15:22

Same theory as Heckle 

. I've been using rspec, heckle and autotest for almost a year now so having a similar set of tools for PHP is a personal itch I'm scratching. Anonymous on Dec 5 2007, 17:34