

## [How To Unit Test Private Methods](#)

By [Roger Keays](#), 19 September 2012

I've discovered that one reason I don't write as many unit tests as I should is because they live in a separate source tree, **out of sight and out of mind**. So I've moved my unit tests to be **inner classes** on the classes they test. A side effect of this is you can easily **test the private methods of a class** since the inner class has direct access to these methods.

Here is the configuration you will need to do this. Note, I am using **TestNG**, but it should be pretty similar in JUnit.

By default [surefire will not execute tests in inner](#)

[classes](#) so first you need to fix this and also tell **maven** where your test code is by adding this to your `<plugins>` section:

```
<!-- unit tests live inside the java source files themselves -->
<plugin>
  <artifactId>maven-surefire-plugin</artifactId>
  <version>2.12</version>
  <configuration>
    <testSourceDirectory>src/main</testSourceDirectory>
    <testClassesDirectory>target/classes</testClassesDirectory>
    <excludes><exclude>[none]</exclude></excludes>
  </configuration>
</plugin>
```

Note that **[none]** is not a special keyword, it only exists to match no class files. You can put whatever rubbish in there (but a blank entry has no effect).

Your unit tests must be **public static inner classes** annotated with **@Test**, like this:

```
@Test
```

```
public static class UserEditorTest {
    public void test_private_method() {
        assertEquals(new UserEditor().somePrivateMethod(), "success");
    }
}
```

*the* **Risk** is **NO**  
**knowing.**  
**Get tested.**

```
}  
}
```

Since the test class is an inner class, it can call the private method.

To run the test from maven, I use a wildcard on the test name so I don't have to specify OuterClass\$InnerClass.

```
$ mvn test -Dtest=*UserEditorTest
```

To get Netbeans to run the tests, you can use a wildcard so it finds the inner class. Go to the project properties and look for the Actions settings. You will need to set the parameters like this:

```
-Dtest={className}
```

You can also run from a normal testng.xml test suite like this:

```
<!DOCTYPE suite SYSTEM "http://testng.org/testng-1.0.dtd">  
<suite name="Suite1" verbose="1">  
  <test name="MyTest" >  
    <classes>  
      <class name="com.example.Example$ExampleTest" />  
    </classes>  
  </test>  
</suite>
```

This setup has improved my productivity and (more importantly) the coverage of my unit tests. Do you think using inner classes for unit testing is a bad idea? Let me know why in the comments below.

## About Roger Keays



Roger Keays is an artist, an engineer, and a student of life. He has no fixed address and has left footprints on 40-something different countries around the world.

Roger is addicted to surfing. His other interests are music, psychology, languages, the proper use of semicolons, and finding good food.