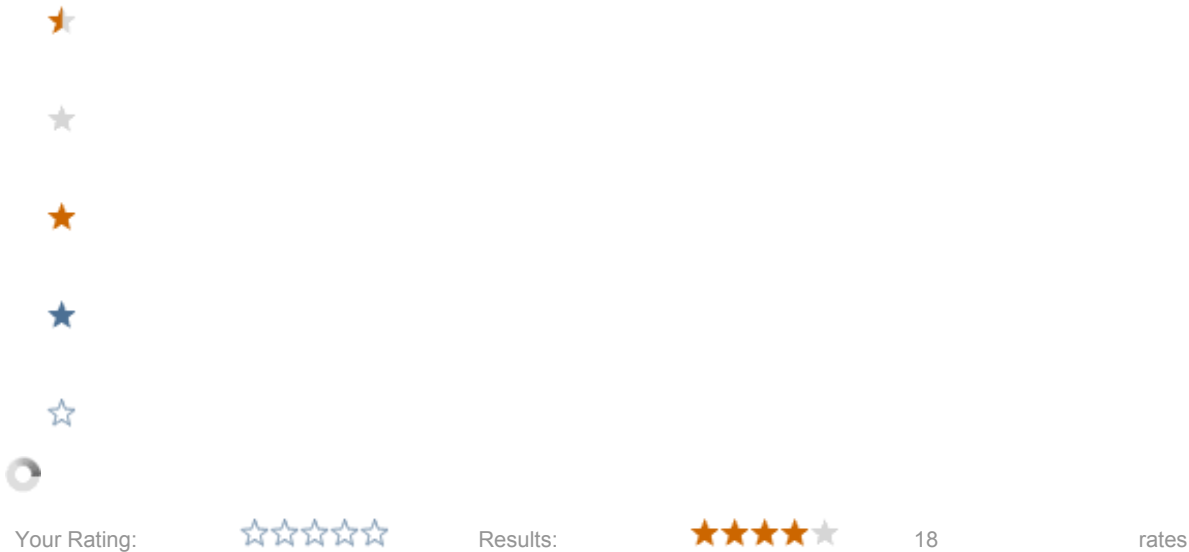


# Converting JUnit3 tests to 4-style



## Intro

Since Magnolia 4.5 we want to use only JUnit4 tests. Most of our existing tests have already been migrated. If you happen to find some JUnit3 tests, you should please convert them.

The bigger part of the conversion can be automated by using the regexps listed beyond. For single test you'll most probably be faster migrating it manually.

## Pitfalls

- JUnit3 allows *setUp* & *tearDown* to be **protected** - after adding the JUnit4 annotations (*@Before*, *@After*) everything will look fine but you'll get an exception at runtime: these must be **public** in JUnit4
- be sure to convert all subclasses of converted tests - else the sub-tests will compile but most likely not run properly as they will be interpreted to be JUnit4-Tests (failures might be shown as success!)
  - indicator for the situation: subclasses show compiler errors on assertX-methods (previously inherited from TestCase, with superclass no longer extending TestCase they are unknown)

## Regexps for automated conversion

The following regexps are taken from [here](#), then adapted to fit our needs - no tabs for indention but spaces:

```
// Add @Test
Replace:
^[ \s]+(public +void +test)
With:
    @Test\n    $1
```

```

Regular Expression: on
Case sensitive: on
File name filter:
*Test.java

// Remove double @Test's on already @Test annotated files
Replace:
^[ \s]*@Test\n[ \s]*@Test
With:
    @Test
Regular Expression: on
Case sensitive: on
File name filter:
*Test.java

// Remove all empty setUp's
Replace:
^[ \s]*+((public|protected) +)?void +setUp\(\)[^\{]*\{\s*(super\.setUp\
(\);)?\s*\}\n([ \s]*\n)?
With nothing
Regular Expression: on
Case sensitive: on
File name filter:
*Test.java

// Add @Before to all setUp's
Replace:
^([ \s]*+@Override\n)?[ \s]*+((public|protected) +)?(void +setUp\(\))
With:
    @Before\n    public void setUp()
Regular Expression: on
Case sensitive: on
File name filter:
*Test.java

// Remove double @Before's on already @Before annotated files
Replace:
^[ \s]*+@Before\n[ \s]*+@Before
With:
    @Before
Regular Expression: on
Case sensitive: on
File name filter:
*Test.java

// Remove all empty tearDown's
Replace:
^[ \s]*+((public|protected) +)?void +tearDown\(\)[^\{]*\{\s*(super\.
tearDown\(\);)?\s*\}\n([ \s]*\n)?
With nothing
Regular Expression: on

```

```
Case sensitive: on
File name filter:
*Test.java

// Add @After to all tearDown's
Replace:
^([\s]+@Override\n)?[\s]+((public|protected) +)?(void +tearDown\(\))
With:
    @After\n    public void tearDown()
Regular Expression: on
Case sensitive: on
File name filter:
*Test.java

// Remove double @After's on already @After annotated files
Replace:
^[ \s]+@After\n[ \s]+@After
With:
    @After
Regular Expression: on
Case sensitive: on
File name filter:
*Test.java

// Remove old imports, add new imports
Replace:
^([\s]*import[\s]+junit\.framework\.Assert;\n)?[\s]*import[\s]+junit\.
framework\.TestCase;
With:
import org.junit.After;\nimport org.junit.Before;\nimport org.junit.Test;
\nimport static org.junit.Assert.*;
Regular Expression: on
Case sensitive: on
File name filter:
*Test.java

// Remove all extends TestCase
Replace:
[\s]+extends[\s]+TestCase[\s]+\{
With:
{
Regular Expression: on
Case sensitive: on
File name filter:
*Test.java

// Look for import junit.framework;
Find:
import junit\.framework
```

```
Manually fix  
Regular Expression: on  
Case sensitive: on
```

```
// Look for ignored tests (FIXME, disabled, ...)
```

```
Find:
```

```
public[ \s]+void[ \s]+\w+test
```

```
Manually fix
```

```
Regular Expression: on
```

```
Case sensitive: on
```

```
// Look for dummy/empty tests
```

```
Find:
```

```
public[ \s]+void[ \s]+test[\w\d]*((\s*)\s*\{(\s*(//[^\n]*)?\s*\}
```

```
Manually fix
```

```
Regular Expression: on
```

```
Case sensitive: on
```