

Simplest JavaBeans Bean and NetBeans, Part 3 Testing Using Junit.



Carl W. David¹
Department of Chemistry
University of Connecticut
Storrs, Connecticut 06269-3060
CarlDavid@uconn.edu

In NetBeans 1.5, while editing beaninput2, under the tools menu, you will find Create JUnit Tests, which will generate a program called beaninput2Test.

It is our task to generate code to augment what's been created here, to automatically test every aspect of our bean (or program, for that matter, or just class). Here is the text with my adjustments highlighted:

```
/*
 * beaninput2Test.java
 * JUnit based test
 *
 * Created on November 29, 2005, 9:41 AM
 */

package input2;

import java.awt.AWTException;
import java.awt.event.KeyEvent;
import junit.framework.*;
import java.awt.Color;
import java.awt.Robot;
import java.beans.PropertyChangeListener;

/**
 *
 * @author david
 */
public class beaninput2Test extends TestCase {
```

¹ As usual, all errors are mine. Please write me with corrections (suggestions, criticisms) at Carl.David@ucon.edu. Thanks.

```

public beaninput2Test(String testName) {
    super(testName);
}

protected void setUp() throws Exception {
}

protected void tearDown() throws Exception {
}

public static Test suite() {
    TestSuite suite = new TestSuite(beaninput2Test.class);

    return suite;
}

/**
 * Test of getProperty method, of class input2.beaninput2.
 */
public void testGetProperty() {
    System.out.println("testgetProperty");

    beaninput2 instance = new beaninput2();

    // TODO add your test code below by replacing the default call to fail.
    //fail("The test case is empty.");
    String expectedResult = "";
    instance.setProperty(expResult);
    String result = instance.getProperty();
    assertEquals(expResult,result);
}

/**
 * Test of getLabel method, of class input2.beaninput2.
 */
public void testGetLabel() {
    System.out.println("testgetLabel");

    beaninput2 instance = new beaninput2();

    // TODO add your test code below by replacing the default call to fail.
    //fail("The test case is empty.");
    String expectedResult = "";
    instance.setLabel(expResult);
    String result = instance.getLabel();
}

```

```

    assertEquals(expResult, result);
}

/**
 * Test of setProperty method, of class input2.beaninput2.
 */
public void testSetProperty() {
    System.out.println("setProperty");

    String value = "";
    beaninput2 instance = new beaninput2();

    instance.setProperty(value);

    // TODO add your test code below by replacing the default call to fail.
    //fail("The test case is empty.");
    //here starts my testing of whether or not this works.
    instance.setProperty("Color.BLUE");
    String returned_Color = instance.getProperty();
    System.out.println(returned_Color);
    assertEquals(instance.getProperty(),"Color.BLUE");
}

/**
 * Test of setLabel method, of class input2.beaninput2.
 */
public void testSetLabel() {
    System.out.println("testsetLabel");

    String newLabel = "";
    beaninput2 instance = new beaninput2();

    instance.setLabel(newLabel);

    // TODO add your test code below by replacing the default call to fail.
    //fail("The test case is empty.");
    assertEquals(instance.getLabel(),newLabel);
}

/**
 * Test of addPropertyChangeListener method, of class input2.beaninput2.
 */
public void testAddPropertyChangeListener() {
    System.out.println("addPropertyChangeListener");

    PropertyChangeListener pcl = null;
    beaninput2 instance = new beaninput2();

```

```

instance.addPropertyChangeListener(pcl);

// TODO add your test code below by replacing the default call to fail.
fail("The test case is empty.");
}

/**
 * Test of removePropertyChangeListener method, of class input2.beaninput2.
 */
public void testRemovePropertyChangeListener() {
    System.out.println("removePropertyChangeListener");

    PropertyChangeListener pcl = null;
    beaninput2 instance = new beaninput2();

    instance.removePropertyChangeListener(pcl);

    // TODO add your test code below by replacing the default call to fail.
    fail("The test case is empty.");
}

/**
 * Test of getColor method, of class input2.beaninput2.
 */
public void testGetColor() {
    System.out.println("getColor");

    beaninput2 instance = new beaninput2();

    // TODO add your test code below by replacing the default call to fail.
    //fail("The test case is empty.");
    Color expectedResult = null;
    instance.setColor(expectedResult);
    Color result = instance.getColor();
    assertEquals(expectedResult, result);
}

/**
 * Test of setColor method, of class input2.beaninput2.
 */
public void testSetColor() {
    System.out.println("testsetColor");

    Color newColor = Color.GREEN;

```

```
beaninput2 instance = new beaninput2();
Color oldColor=instance.getColor();
instance.setColor(newColor);

// TODO add your test code below by replacing the default call to fail.
//fail("The test case is empty.");
assertEquals(instance.getColor(),newColor);
}
}
```

Unsophisticated but effective, this scheme allows testing unintrusively of our bean code. This could, and most likely should be improved, but the idea here is to just illustrate the tool rather than discuss debugging and error discovery techniques.