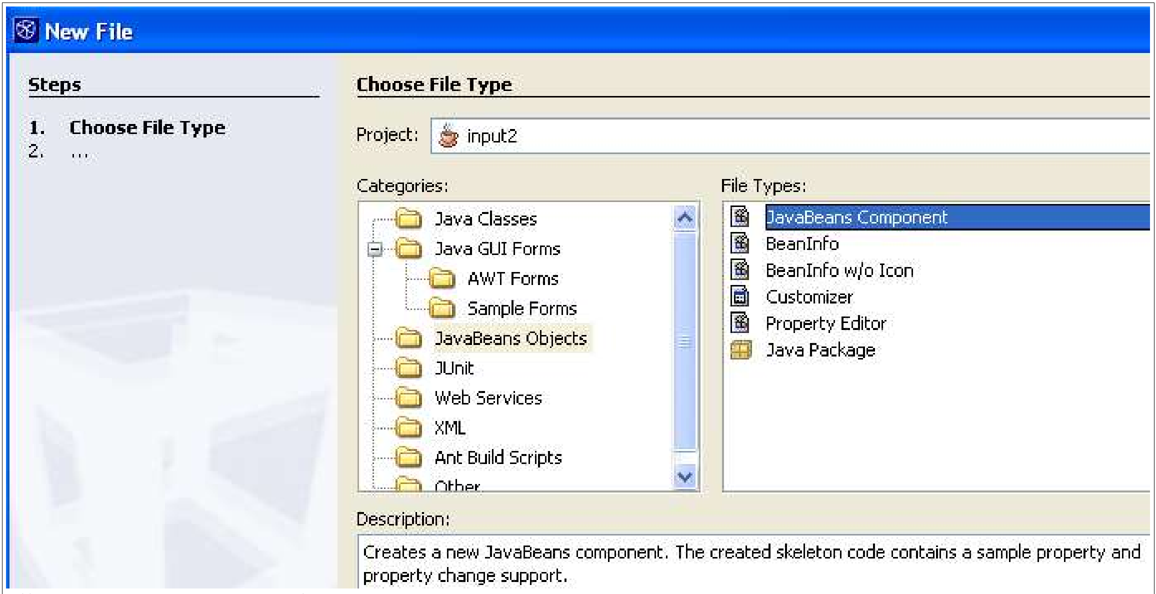
Simplest Do Nothing JavaBean in NetBeans

Carl W. David
Department of Chemistry
University of Connecticut
Storrs, Connecticut 06269-3060[1]
CarlDavid@uconn.edu

Our goal is to create the simplest JavaBean bean and then to create the simplest application which can use this bean. Call it the hydrogen atom (chemists and physicists will understand this usage) of JavaBean technology. Our first bean does nothing, I repeat, nothing, but displays a button. Big deal.

To generate a Bean, once one has to define a project; then one does New File/JavaBean Objects/JavaBean Component:

1 November, 2006; NetBeans 1.5, Java 1.5

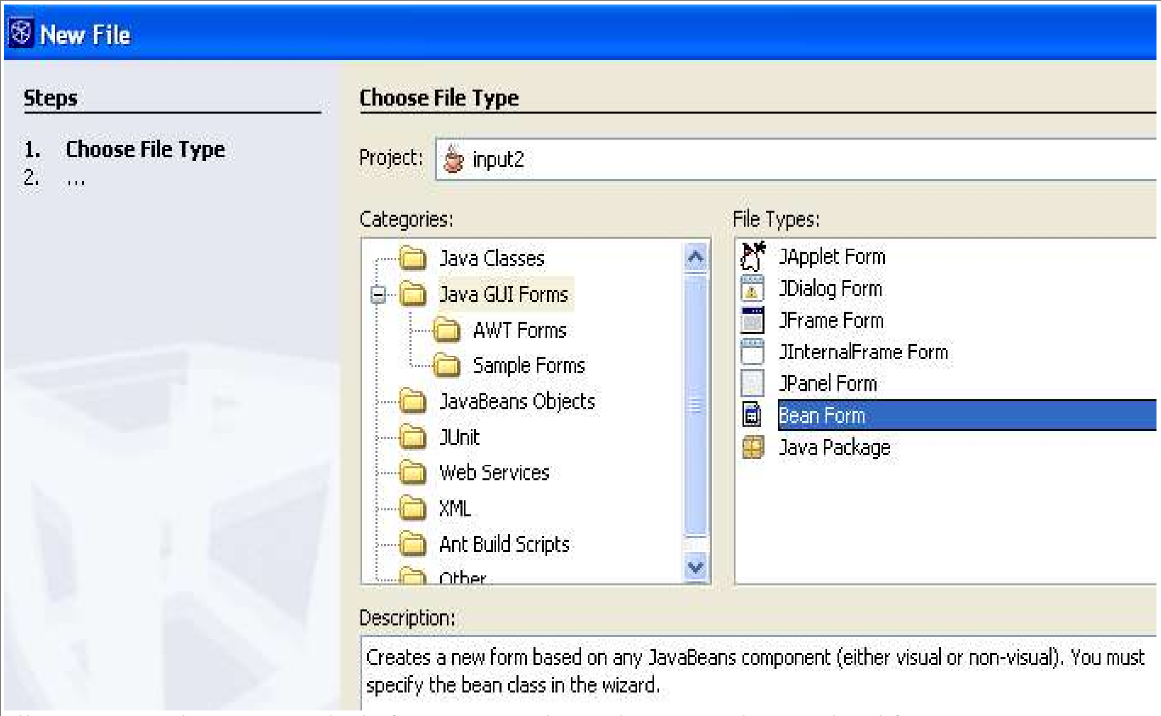


Creates a new JavaBeans component. The created skeleton code contains a sample property and property change support.

Illustration 1: Setting up a bean

and giving it a unique package name ("inp", in our case), so that we can refer to it later outside the development arena.

Instead, to obtain a JPanel based bean, use:



Creates a new form based on any JavaBeans component (either visual or non-visual). You must specify the bean class in the wizard.

Illustration 2: Alternative method of generating a bean, this time with a graphical face

and then choose javax.swing.JPanel when the wizard asks for a superclass. This latter method is the one used here.

Application Tester

No matter what we do in the bean, we need an Application (or Applet) to exercise it. So, we create one. It is so simple that we list it here in its entirety.

```
/*
 * Main.java
 *
 * Created on November 7, 2005, 1:42 PM
 *
 * To change this template, choose Tools | Template Manager
 * and open the template in the editor.
 */

package input2;


import javax.swing.JFrame;
import inp.beaninput2;
/**
 *
 * @author david
 */
public class Main {

    /** Creates a new instance of Main */
    public Main() {
    }

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        // TODO code application logic here
        beaninput2 myBean = new beaninput2();
        JFrame myFrame = new JFrame();
        myFrame.add(myBean);//this worked
      //myFrame.getContentPane().add(myBean);//this worked too
        myFrame.setSize(600,600);
        myFrame.setVisible(true);
    }

}
```

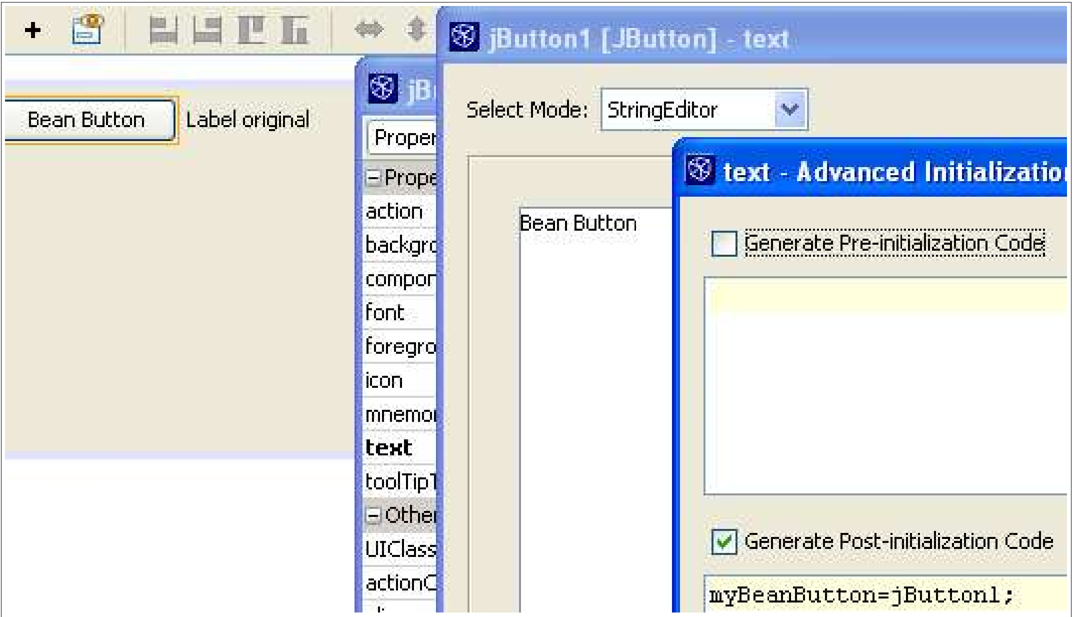
There are other ways of doing this, but for the time being this is good enough. Note the code:

```
import inp.beaninput2;
```

which is why we gave the bean its own package. Then we create the bean, add it to the just created JFrame, set the JFrame's size, and show it. This is one technique for seeing what's going on and debugging it.

Our Do Nothing Bean (good for nothing!)

The bean itself couldn't be simpler, we just dragged a JButton from the pallete, made its name public " myBeanButton=jButton1;" in post-initialization code, changed its text, did the same for a JLabel and that was that. You can't see the generated code because, to save space, I've deleted it. In the Design view, however, it looked like:



```
myBeanButton=jButton1;
```

Illustration 3: in Design mode, we've edited the properties of the button, adding myBeanButton as a publically accessible name

Here is the editable code for the bean:

```
/*
 * beaniput2.java
 *
 * Created on November 8, 2005, 11:00 AM
 */

package inp;

/**
 *
 * @author  david
 */
public class beaninput2 extends javax.swing.JPanel {
  /** Creates new form BeanForm */
  public beaninput2() {
     initComponents();
     System.out.println("bean initialized?");
     this.setVisible(true);
  }

  /** This method is called from within the constructor to
   * initialize the form.
   * WARNING: Do NOT modify this code. The content of this method is
   * always regenerated by the Form Editor.
   */
  // internal editor-fold material deleted to save space!


  // Variables declaration - do not modify
  // End of variables declaration
  public static JButton myBeanButton;
  pubic JLabel myLabel;
}
```

There is nothing for this bean to do, i.e., it just sits there and shows a button and a label. Nothing could be simpler (I think). To repeat, you can see in Illustration 3, the Visual Design of the Bean (on the l.h.s. of the figure). It consists of a button and an label, that's all. The initialization code that creates it is hidden in the un-editable part of the source code, commented out in the above listing.In the next part of this discussion, we will have the bean do something!