JavaBean Examples

How to make a simple java bean using NetBeans

Step 1 - Create a library

From a fresh opening of NetBeans, do:

1. file->new project
2. for category choose 'javaAnt'
   for project choose 'java class library'
   Then choose an appropriate location and give name to your library.

Step 2 - Create an Empty class

1. file->new file
   for category choose-- 'java'
   for file type choose --'Java Class
   click 'next'
   Enter an appropriate class name.
   Leave everything else untouched.

Step 3 - Make code changes

1. Then write your JavaBean code in class which is created by you and compile the program

For example:

```
import java.awt.*;
import java.io.Serializable;
public class Bean extends Canvas implements Serializable
{
public Bean ()
{
setSize(70,50);
setBackground(Color.red);
}
```

}

Step 4 - Fix manifest to show this is a bean library

1. First get into the files tab of your project.
2. In the top level of the tree view, it shows your projects.
3. The next level down should have a file called build.xml.
4. Click the plus sign next to it. Then double click any element below build.xml.
5. You should be now editing a file called build-impl.xml.
6. Search for "<manifest" (I include the open < to simplify the search).
7. There should only be one such tag in the file and you should be in the jar building section (scroll up to see the next comment block if you want to double check).
8. There should already be 2 attributes in the manifest. Add one called Java-Bean as shown:

```
<manifest>
   <attribute name="Main-Class" value="${main.class}"/>
   <attribute name="Class-Path" value="${jar.classpath}"/>
   <attribute name="Java-Bean" value="true"/>
</manifest>
```

9. Now go back to projects and do the following!

Step 5 – Create JFrameForm

1. Right click on package
2. And click on new
3. And click on JFrameForm
4. Give any class name for JFrame

Step 6 - Test and Debug

1. In menu system do: tools -> palette -> Swing/AWT Components

2. Click on Add From Project...
   Navigate and choose your library project created in step 1. Click next.
   Your component should show up. Click it and click 'next' again.
   Select a folder to put your bean into - "Beans". Click 'finish'.
3. Now you can see your bean in the pallette near the bottom in the Beans section. Open up the Beans section if it isn't already.
4. Click your bean and then click in your window to place it.
5. Now right click on JFrame.java and click on “Debug File”.
6. Now right click on JFrame.java and click on “Run File”.

Program-1

Aim: Java program to create a JavaBean to display a red color rectangle

```
import java.awt.*;
import java.io.Serializable;
public class RectangleBean extends Canvas implements Serializable
{
public RectangleBean()
{
setSize(70,50);
setBackground(Color.red);
}
}
```

Program-2

Aim: JavaBean program to change the color of bean based on the mouse press (ref :: java complete reference book)

a Bean called Colors, shown here

```
// A simple Bean.
import java.awt.*;
import java.awt.event.*;
import java.io.Serializable;
public class Colors extends Canvas implements Serializable
{
transient private Color color; // not persistent
private boolean rectangular; // is persistent
public Colors()
{
addMouseListener(new MouseAdapter()
{
public void mousePressed(MouseEvent me)
{
change();
}
});
rectangular = false;
setSize(200, 100);
change();
}


public boolean getRectangular()
{
return rectangular;
}
public void setRectangular(boolean flag)
{
this.rectangular = flag;
```

```
repaint();
}
public void change()
 {
color = randomColor();
repaint();
}
private Color randomColor()
 {
int r = (int)(255*Math.random());
int g = (int)(255*Math.random());
int b = (int)(255*Math.random());
return new Color(r, g, b);
}
public void paint(Graphics g)
{
Dimension d = getSize();
int h = d.height;
int w = d.width;
g.setColor(color);
if(rectangular)
{
g.fillRect(0, 0, w-1, h-1);
}
else
 {
g.fillOval(0, 0, w-1, h-1);
}
}

}
```

Program-3

Aim: JavaBean program to introspect a Bean

```
//Create a FactBean
import java.io.*;
public class  FactBean implements Serializable
{
protected int n, fact=1;
public FactBean(){   }
public int getN()
{
return n;
}
public void setN(int n)
{
 this.n=n;
for(int i=1;i<=n;i++)
        {       fact=fact*i;
        }
}
public int getFact()
{
return fact;
}
}
```

//Introspect FactBean :

```
import java.beans.*;
public class  IntroBean
{
public static void main(String args[])throws IntrospectionException
{
BeanInfo info=Introspector.getBeanInfo(FactBean.class,Object.class);
for(PropertyDescriptor pd:info.getPropertyDescriptors())
{
System.out.println("Bean Properties:=" + pd.getName());
```

```
}
for (MethodDescriptor md : info.getMethodDescriptors())
{
System.out.println(" Bean Methods: " + md.getName());
}
}
}
```