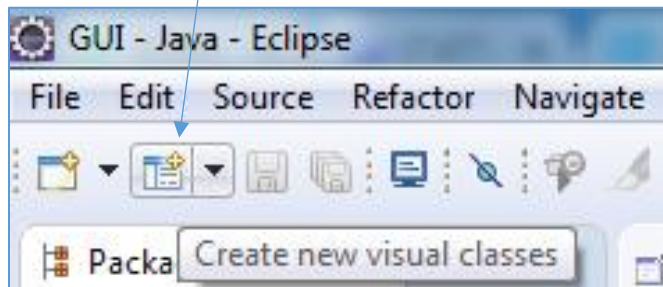


## Java WindowBuilder GUI

Java AWT **Abstract Windowing Toolkit** the original GUI toolkit shipped with the Java Development Kit (JDK). Swing: latest GUI toolkit. AWT heavyweight components: depend on native libraries.

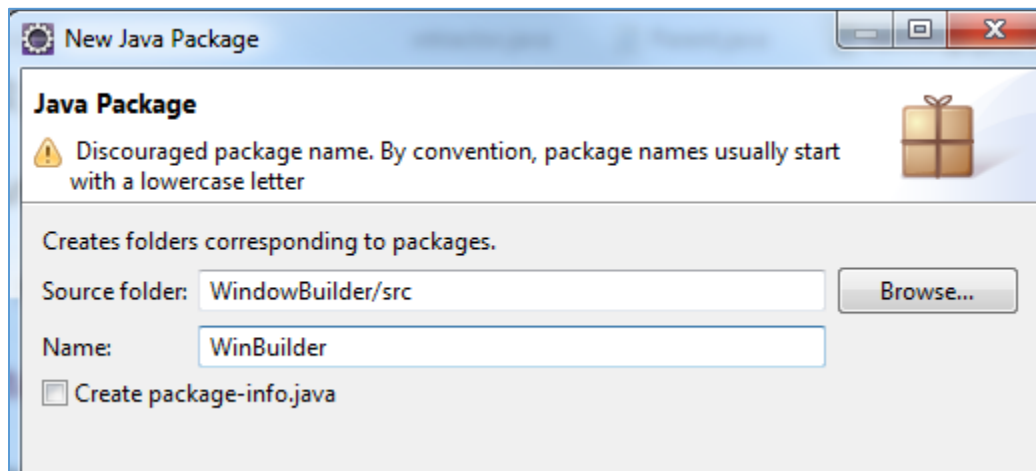
Check the second icon



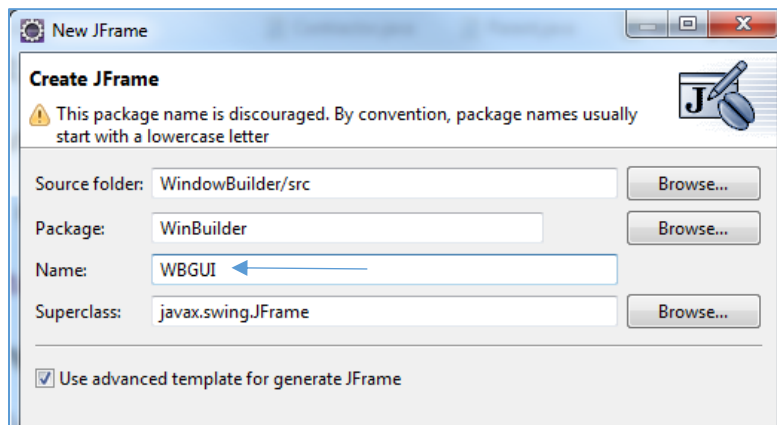
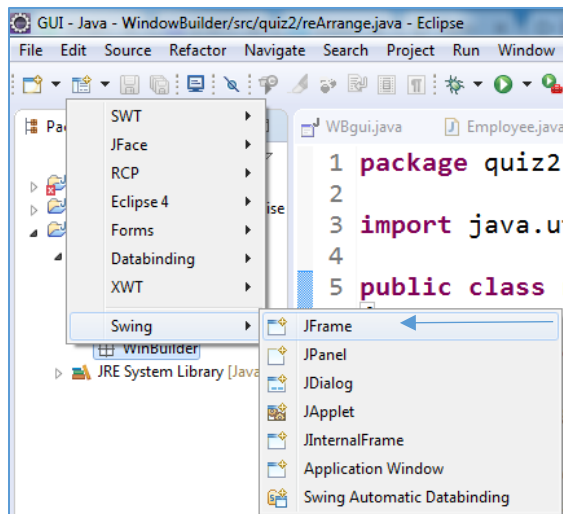
If the icon is not there, follow the install instructions to install WindowBuilder.

Create a Java project called WindowBuilder

Create a Java package called WinBuilder



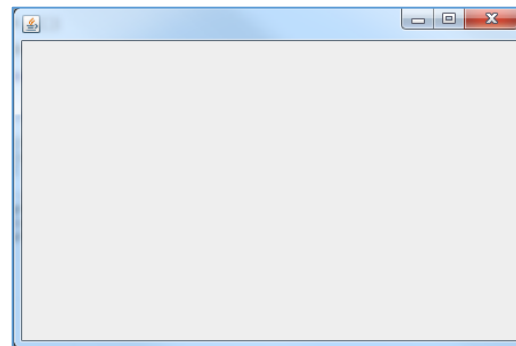
Click on the WindowBuilder icon, Swing, JFrame



type in WBGUI for the JFrame name

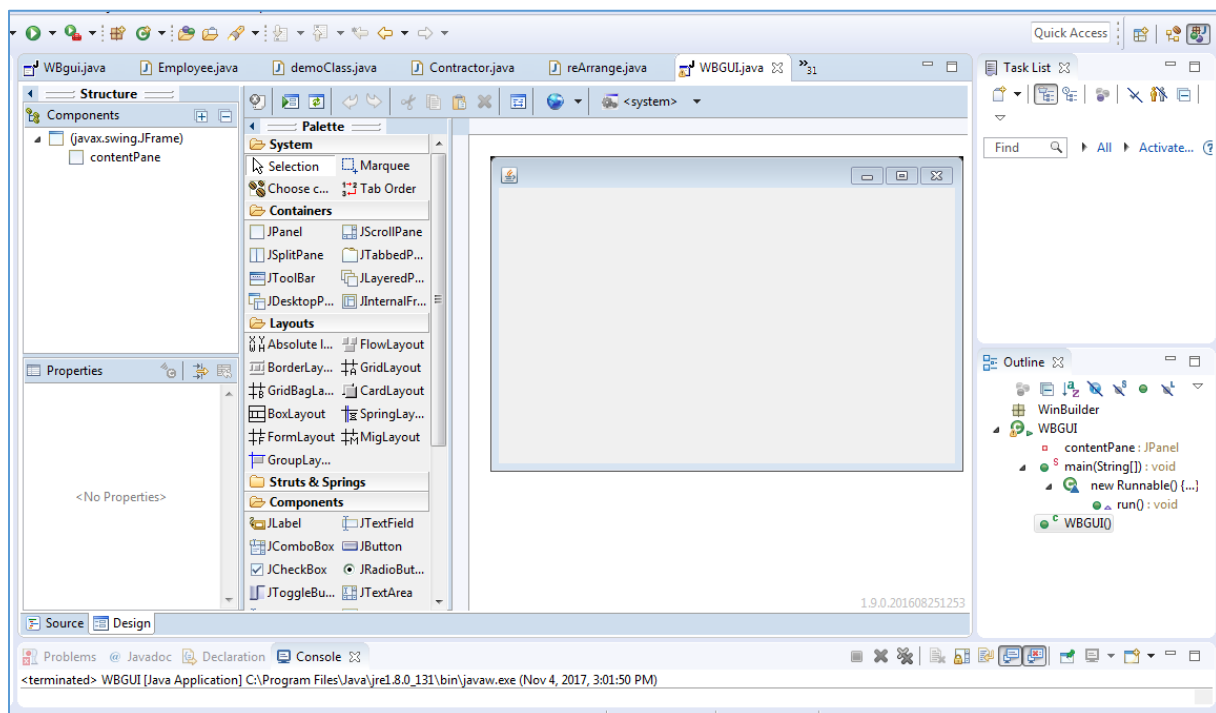
Eclipse will present you the source code for this frame. Run it. Your line # may be different.

```
1 package WinBuilder;
2
3 import java.awt.BorderLayout;
4
5
6
7
8
9
10 public class WBGUI extends JFrame {
11
12     private JPanel contentPane;
13
14     /**
15      * Launch the application.
16      */
17     public static void main(String[] args) {
18         EventQueue.invokeLater(new Runnable() {
19             public void run() {
20                 try {
21                     WBGUI frame = new WBGUI();
22                     frame.setVisible(true);
23                 } catch (Exception e) {
24                     e.printStackTrace();
25                 }
26             }
27         });
28     }
29
30     /**
31      * Create the frame.
32      */
33     public WBGUI() {
34         setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
35         setBounds(100, 100, 450, 300);
36         contentPane = new JPanel();
37         contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
38         contentPane.setLayout(new BorderLayout(0, 0));
39         setContentPane(contentPane);
40     }
41 }
42
```

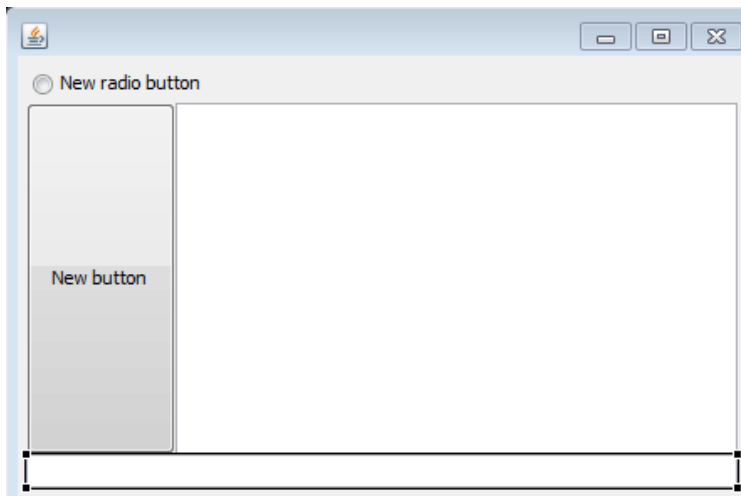


Look at at 21 and 22.

Tap on the Design tab. It may take a while before you can see the design screen as shown on the next page.



Try to put some components onto your frame.



Now try to remove them.

Add a button to the frame and switch back to the Source screen.

You may have something similar to Lines 44 and 45 added (see next page)

```
33-  /**
34-   * Create the frame.
35-   */
36-  public WBGUI() {
37-      setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
38-      setBounds(100, 100, 450, 300);
39-      contentPane = new JPanel();
40-      contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
41-      contentPane.setLayout(new BorderLayout(0, 0));
42-      setContentPane(contentPane);
43-
44-      JButton btnNewButton = new JButton("New button");
45-      contentPane.add(btnNewButton, BorderLayout.WEST);
46-  }
```

Delete these two lines and switch back to the Design.

Back to the Source. Create a method called `setupComponents` and move the code inside `WBGUI` into it.

```

33 private void setupComponents()
34 {
35     setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
36     setBounds(100, 100, 450, 300);
37     contentPane = new JPanel();
38     contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
39     contentPane.setLayout(new BorderLayout(0, 0));
40     setContentPane(contentPane);
41 }
42 /**
43  * Create the frame.
44  */
45 public WBGUI() {
46     setupComponents();
47 }
48
49 }

```

Now switch back to design and add a button to the left.

```

33 private void setupComponents()
34 {
35     setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
36     setBounds(100, 100, 450, 300);
37     contentPane = new JPanel();
38     contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
39     contentPane.setLayout(new BorderLayout(0, 0));
40     setContentPane(contentPane);
41
42     JButton btnNewButton = new JButton("New button");
43     contentPane.add(btnNewButton, BorderLayout.WEST);
44 }
45 /**
46  * Create the frame.
47  */
48 public WBGUI() {
49     setupComponents();
50 }
51 }

```

Eclipse automatically puts the two required lines (42 43) of code into the setupComponents method.

Add one more button to the frame

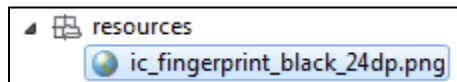
Look at the Properties area and make some changes

Go to

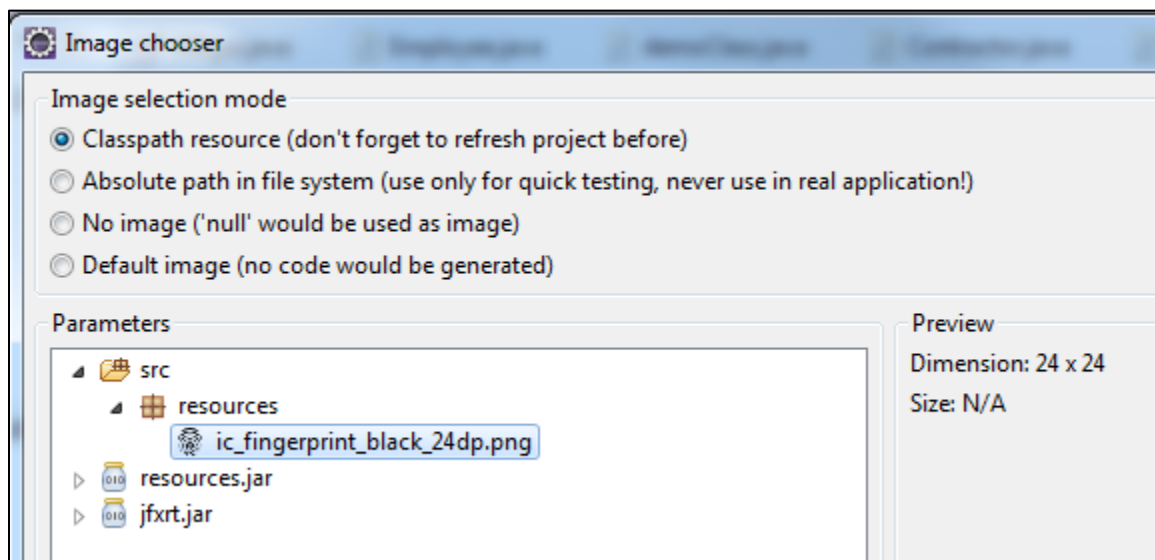
[https://material.io/icons/?utm\\_source=google&utm\\_medium=cpc&utm\\_campaign=1001467%20%7C%20Material.IO%20%7C%20Global%20%7C%20en%20%7C%20Hybrid%20%7C%20Text%20%7C%20SKWS&utm\\_term=%7Bkeyword%7D&gclid=Cj0KCQjwyvXPBRD-ARIsAleQeoHWq8OSPzu88DnBCErDLLQjLMOQ40OE918fSTdq-qqClUO42y0Nx3UaAqg9EALw\\_wcB](https://material.io/icons/?utm_source=google&utm_medium=cpc&utm_campaign=1001467%20%7C%20Material.IO%20%7C%20Global%20%7C%20en%20%7C%20Hybrid%20%7C%20Text%20%7C%20SKWS&utm_term=%7Bkeyword%7D&gclid=Cj0KCQjwyvXPBRD-ARIsAleQeoHWq8OSPzu88DnBCErDLLQjLMOQ40OE918fSTdq-qqClUO42y0Nx3UaAqg9EALw_wcB)

and download some icons

Create a package called resources and put the icons there

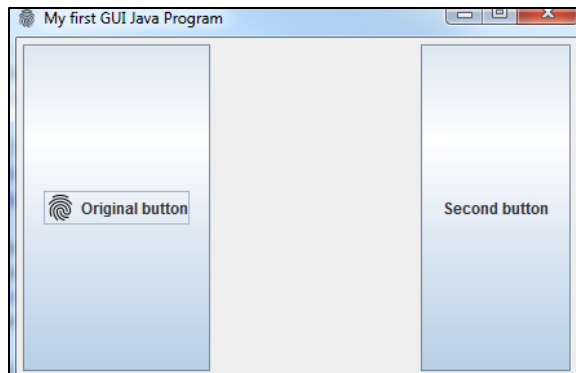


Select your left button and then the icon property ...



Do the same for the frame iconimage

You can change the name of the button either on the Design screen or Source screen.



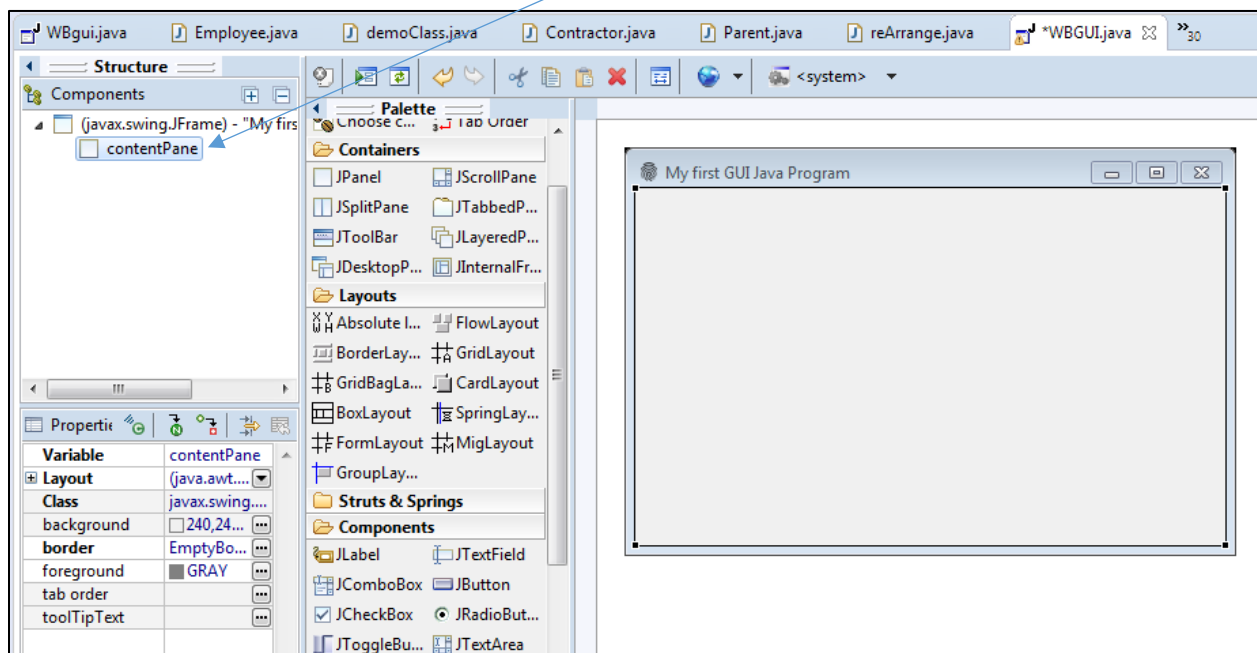
```

41 private void setUpComponents()
42 {
43     setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
44     setBounds(100, 100, 450, 300);
45     contentPane = new JPanel();
46     contentPane.setForeground(Color.GRAY);
47     contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
48     contentPane.setLayout(new BorderLayout(0, 0));
49     setContentPane(contentPane);
50     setIconImage(Toolkit.getDefaultToolkit().getImage(WBGUI.class
51         .getResource("/resources/ic_fingerprint_black_24dp.png")));
52     setTitle("My first GUI Java Program");
53
54     JButton btnOriginal = new JButton("Original button");
55     btnOriginal.setIcon(new ImageIcon(WBGUI.class
56         .getResource("/resources/ic_fingerprint_black_24dp.png")));
57     contentPane.add(btnOriginal, BorderLayout.WEST);
58
59     JButton btnSecond = new JButton("Second button");
60     contentPane.add(btnSecond, BorderLayout.EAST);
61 }

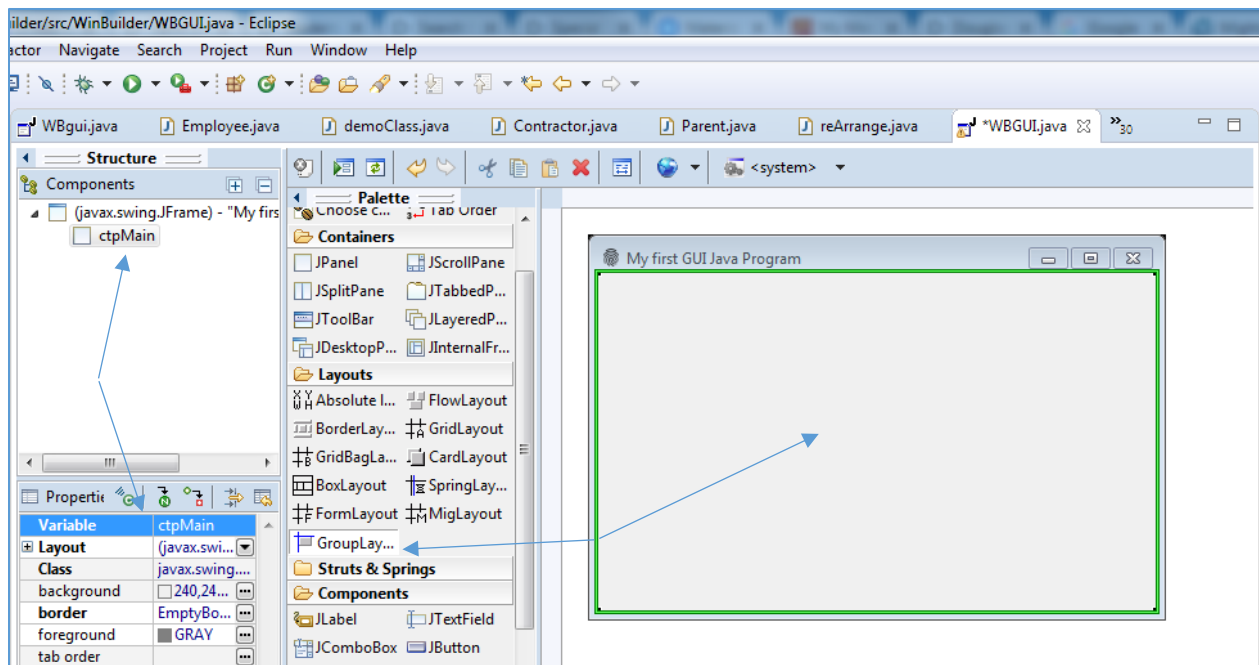
```

Try to delete 50 51 and see the difference

Now delete the two buttons and click on contentPane



Change it to ctpMain. On the Laoyouts, click on GroupLayout, drop it to the ctpMain

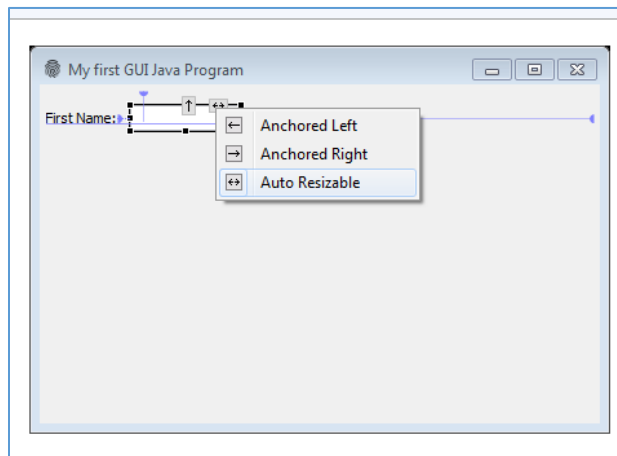


Now drag and drop a button the ctpMain. You are free to drop it anywhere unlike what you had on page 3. You can also change the size of the button easily. Change the icon of this new button.







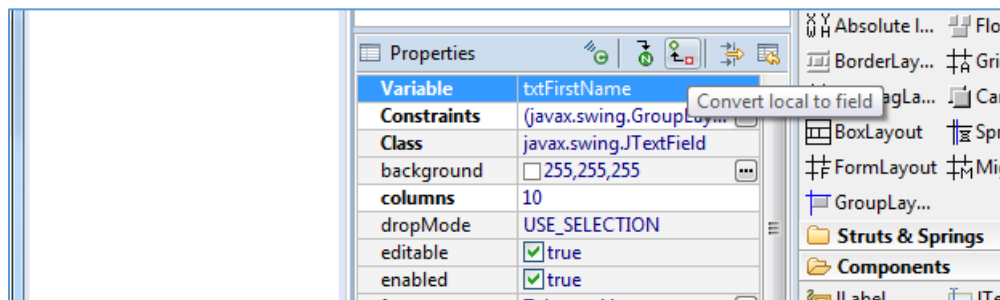
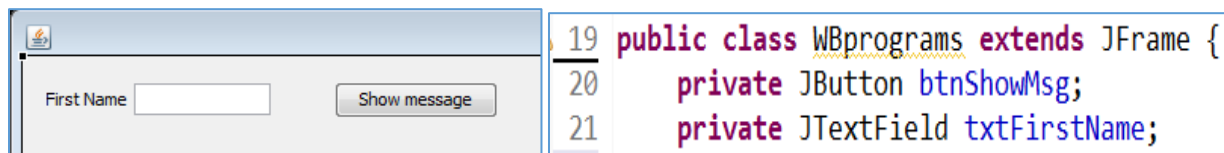
Delete a component from the Design screen is much easier and cleaner than on the Source screen.



Now delete the button and add: JLabel, JTextField

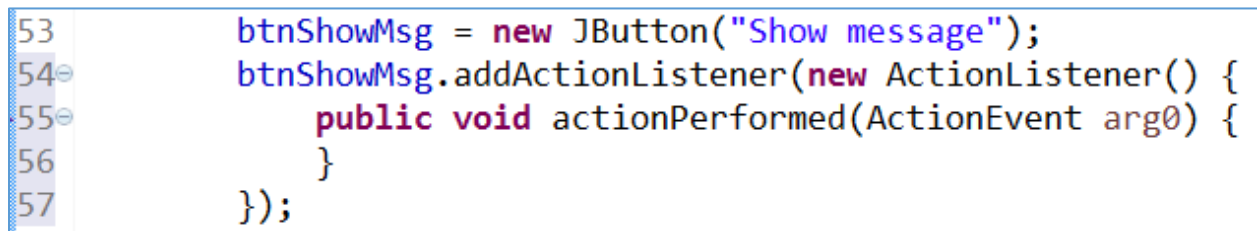
Try out the Auto Resizable to see how it works

Create a JLabel, a JTextField, and a JButton as shown below. Select the JTextField and change the Variable to txtFirstName. Change the button to btnShowMsg. Click on the icon  to convert the text field  for both objects. If you switch back to Source, you have the following:



Now double click on the message button. You open up Source automatically.

Lines 54-57 is a single statement: button's ActionListener



Now create a createEvents method in Source and move 54-57 to this method.

```

41 private void createEvents()
42 {
43     btnShowMsg.addActionListener(new ActionListener() {
44         public void actionPerformed(ActionEvent arg0) {
45         }
46     });
47 }

```

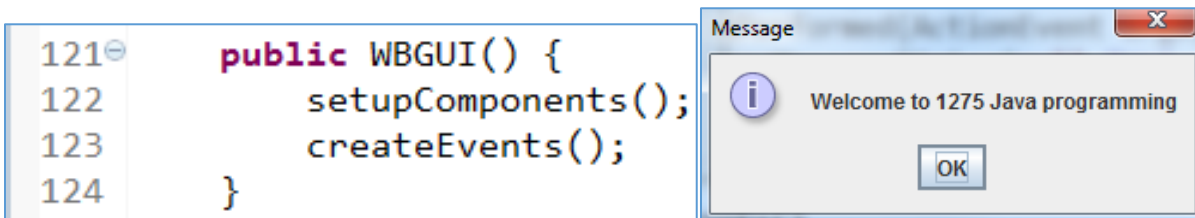
Add Line 45 as shown

```

41 private void createEvents()
42 {
43     btnShowMsg.addActionListener(new ActionListener() {
44         public void actionPerformed(ActionEvent arg0) {
45             JOptionPane.showMessageDialog(null, "Welcome to 1275 Java programming");
46         }
47     });
48 }

```

Add line 123 as shown. Run your program and click on the button to see the message.



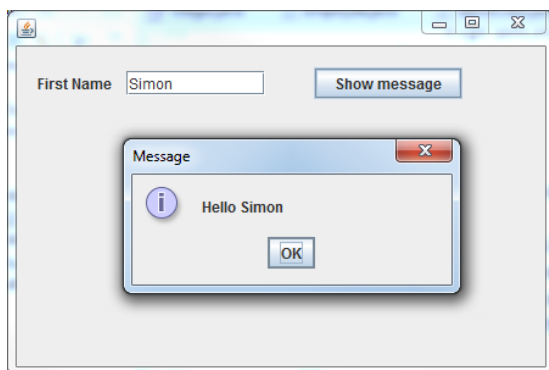
Change Line 45 to the following

```

45     JOptionPane.showMessageDialog(null, "Hello "
46                                     + txtFirstName.getText());

```

txtFirstName.getText() is to \_\_\_\_\_. Now run your program again.



Let's create a new project (Chapter14) new package (demoGUI)

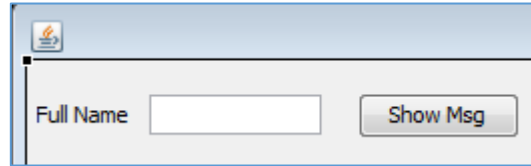
Let's create another JFrame (wbDemo) from Swing

Add a GroupLayout, JLabel, JTextField (txtName), JButton (btnShowMsg)

Convert txtName and btnShowMsg to field

Click on the button to go the Source

Create the two methods below



```
public void createEvents()
{

}
public void setupComponents()
{

}
```

Move the wbDemo code to setupComponents.

Move the following lines to createEvents

```
btnShowMsg.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent arg0) {
    }
});
```

If you switch back to the Design screen, all components are gone, why?

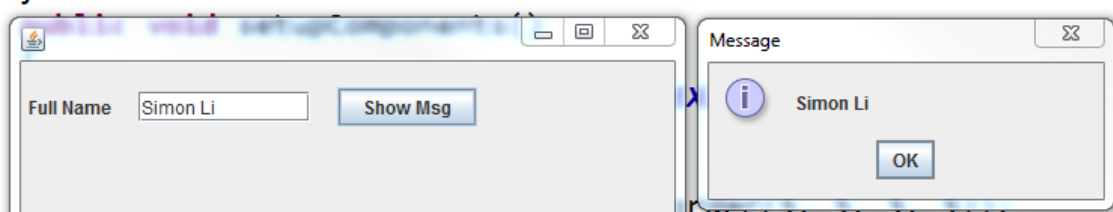
Enter 90 91 into wbDemo()

```
88 public wbDemo()
89 {
90     setupComponents();
91     createEvents();
92 }
```

Switch back to the Design screen

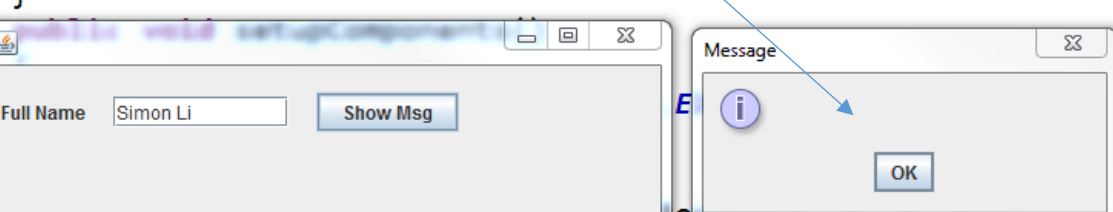
Double click on the btnShowMsg

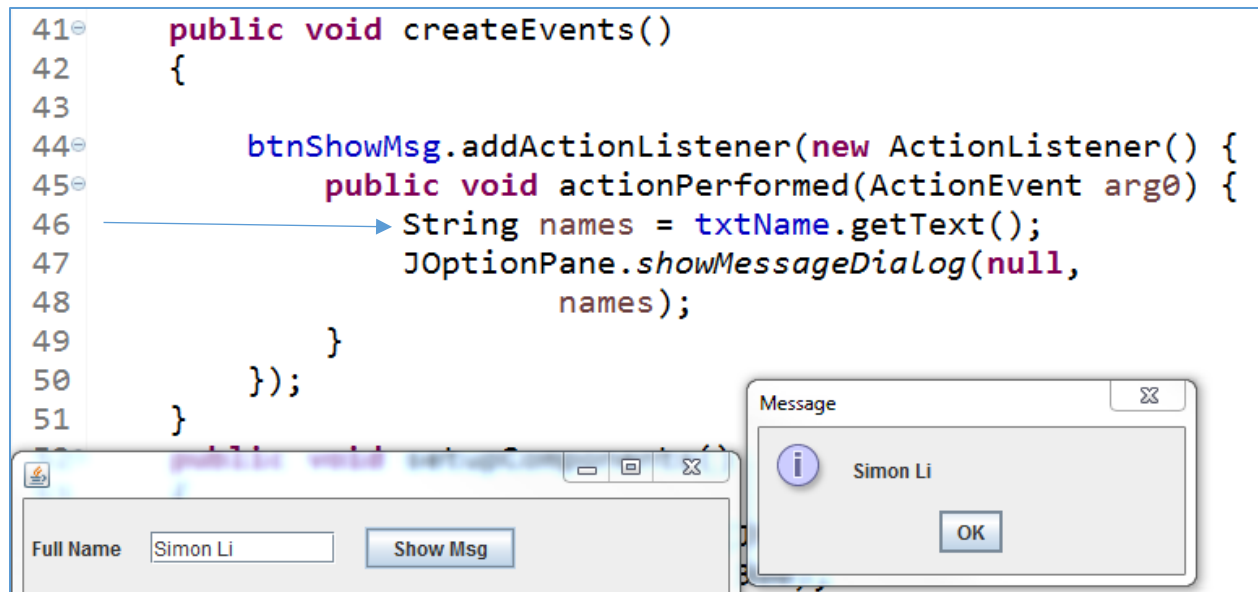
```
41 public void createEvents()
42 {
43
44     btnShowMsg.addActionListener(new ActionListener() {
45         public void actionPerformed(ActionEvent arg0) {
46             JOptionPane.showMessageDialog(null,
47                 txtName.getText());
48         }
49     });
50 }
51
52
53
54
55
56
```



So far so good but why the following?

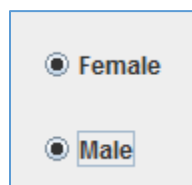
```
41 public void createEvents()
42 {
43     String names = txtName.getText();
44     btnShowMsg.addActionListener(new ActionListener() {
45         public void actionPerformed(ActionEvent arg0) {
46             JOptionPane.showMessageDialog(null,
47                 names);
48         }
49     });
50 }
51
52
53
54
55
56
```





Now change your code just to display the first name

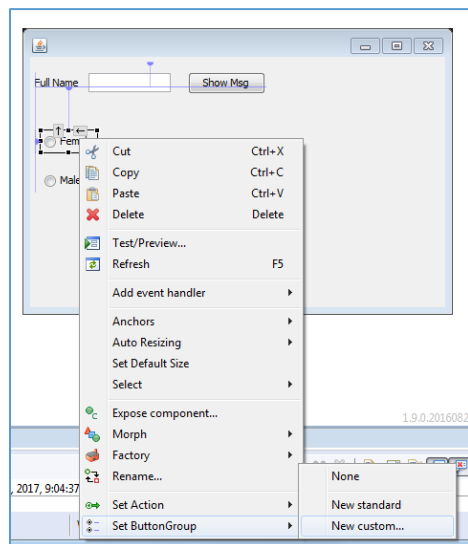
Add two JRadioButtons (rdbtnFemale, rdbtnMale). Make sure that they are fields, not local.



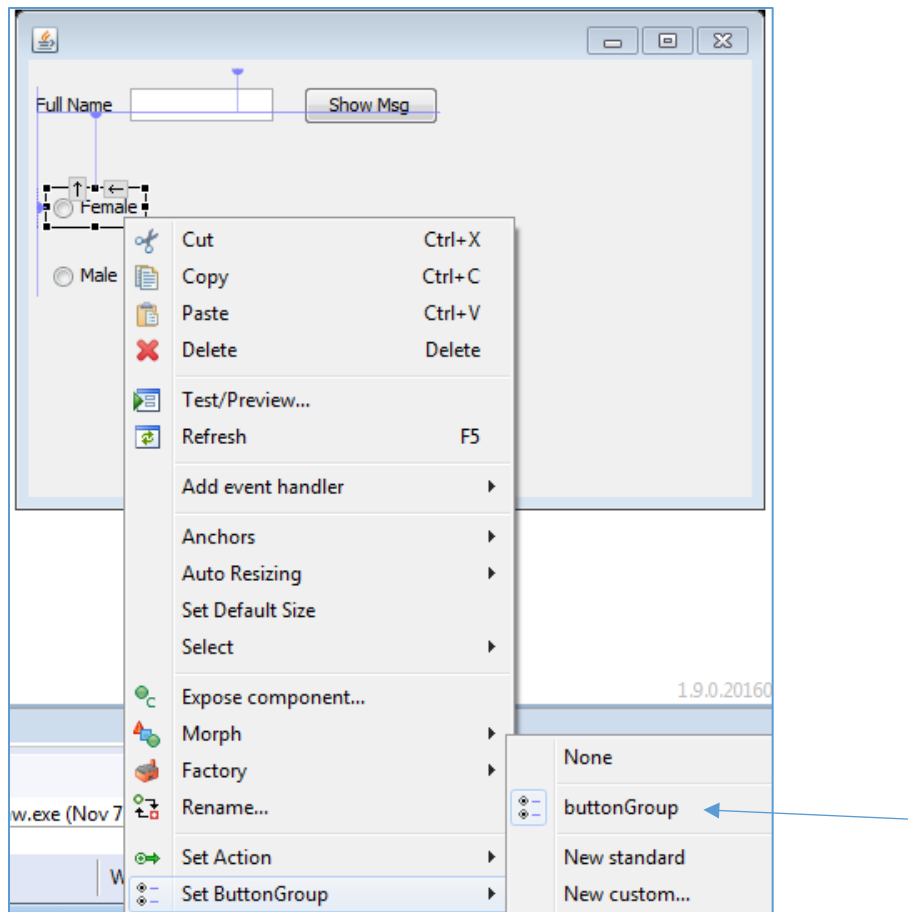
the radio buttons are not mutually exclusive.

Right click anywhere on the frame

Right click on one of the radio buttons



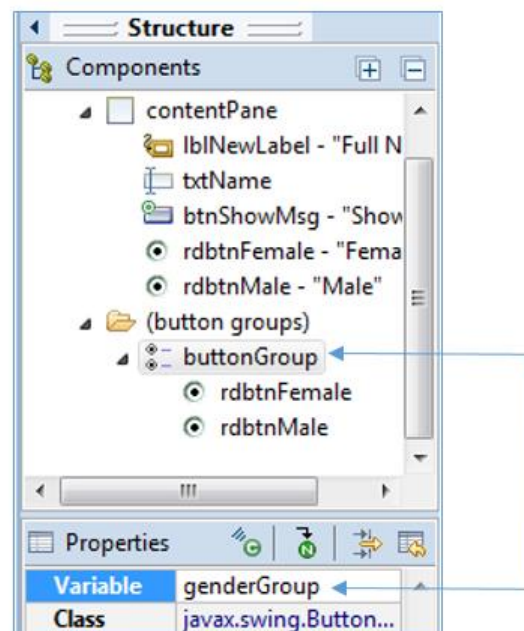
select New standard.



You now have a buttonGroup.

Right click on each radio button and select buttonGroup

You may want to change the group name to some more meaningful, such as genderGroup.



```

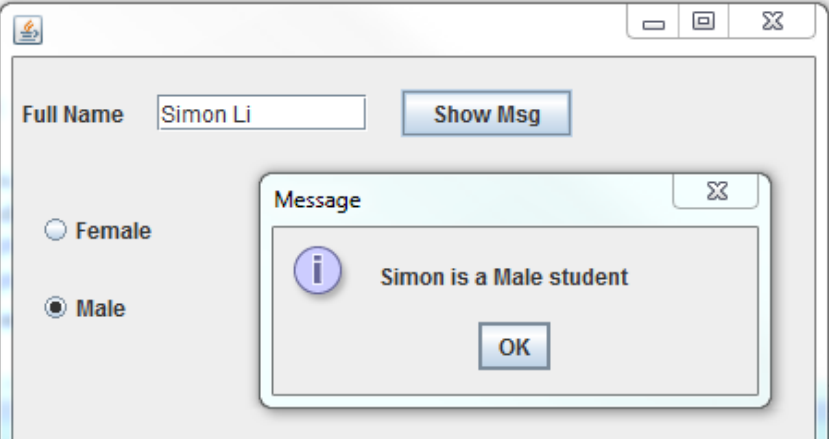
77         rdbtnFemale = new JRadioButton("Female");
78         genderGroup.add(rdbtnFemale);
79
80         rdbtnMale = new JRadioButton("Male");
81         genderGroup.add(rdbtnMale);

```

```

46 public void createEvents()
47 {
48
49     btnShowMsg.addActionListener(new ActionListener() {
50     public void actionPerformed(ActionEvent arg0) {
51         String gender;
52         if (rdbtnFemale.isSelected())
53             gender = "Female";
54         else
55             gender = "Male";
56         String[] names = txtName.getText().split(" ");
57         JOptionPane.showMessageDialog(null,
58             names[0] + " is a " + gender + " student");
59     }
60 });
61 }
62 public void
63 {
64     setDefa
65     setBoun
66     content
67     content
68     setCont
69
70

```



Now go back to AS1 and re-code your program using WindowBuider GUI.