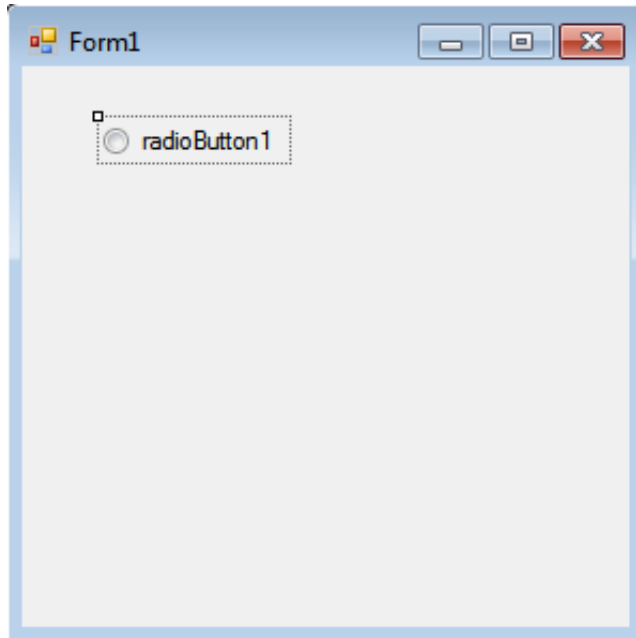
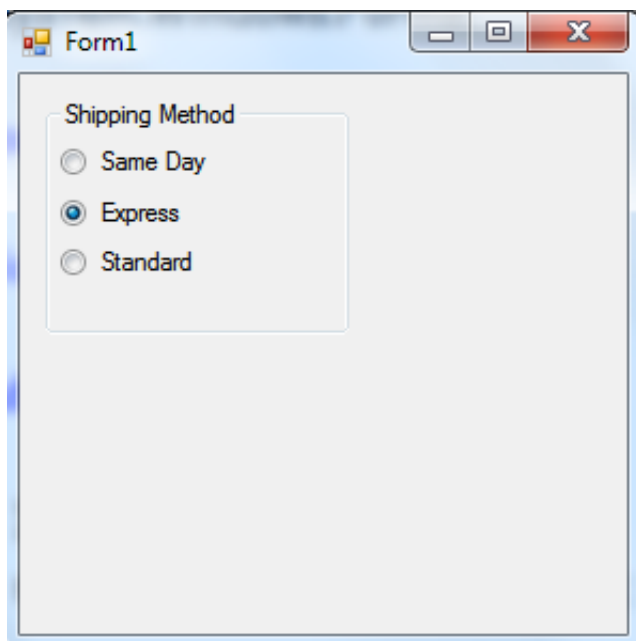


Radio Buttons

Radio Buttons allow the user to choose one item from a group of several choices. When you select a Radio Button, any previously selected Button becomes un-selected. This means that the choices are mutually exclusive, i.e. you can only ever choose one Radio Button from the group. If you want to be able to choose several items from a group Visual Studio provides a CheckBox control to do this.



The figure here shows a group of Radio Buttons on a Form. A Radio Button appears as a small circle with some text to the right of the circle. The text usually explains what the Radio Button represents. It is stored in the Text property of the Radio Button. When a Radio Button is selected, a small dot appears inside of the circle.



Radio Buttons always appear in a group. A group is defined by whatever enclosing control holds the radio buttons. By default this is the Form, but you can also use a Panel or a GroupBox control. This figure shows a group of three Radio Buttons inside of a GroupBox. The Group Box provides a Text property that helps describe what the Radio Buttons are for. In this

case the Text property of the Group Box has been changed to "Shipping Method". The middle Radio Button has been selected.

If you have a Radio Button named `radioBtn1` you can get the text of the selected item by writing

```
string selectedItem = radioBtn1.Text;
```

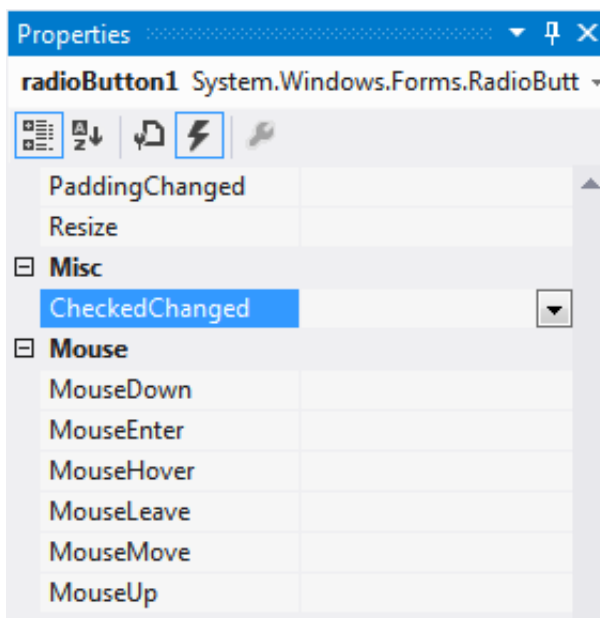
You can test a Radio Button to see whether or not it has been selected (checked) by looking at its `Checked` property. If the Radio Button is the currently selected Button in the group, its `Checked` property will be `true`. Otherwise it will be `false`. For example,

```
if (radioBtn1.Checked) { ... }
```

You can set the `Checked` property of a Radio Button by writing

```
radioBtn1.Checked = true;
```

Handling the `CheckedChanged` Event



When you select or de-select a Radio Button, a `CheckedChanged` event is generated. Create the skeleton for a `CheckedChanged` event handler by selecting a Radio Button. Then in the properties window click on the lightning bolt to see a list of possible events. Double click on `CheckedChanged`.

Here is an example of a `CheckedChanged` event handler:

```
private void radioButton1_CheckedChanged(object sender, EventArgs e)
{
    if (radioButton2.Checked)
        MessageBox.Show("selected Express
```

```
Shipping");  
}
```