How do I make an Event in the Usercontrol and have it handled in the Main Form?

Ask Question



I have a custom usercontrol and I want to do something relatively simple.

55

When ever a numeric up down in that usercontrol's value changes, have the main form update a display window.



This is not a problem if the NUD was not in a usercontrol but I can't seem to figure out how to have the event handled by the mainform and not the usercontrol.



1 possible duplicate of <u>Winforms user controls custom events</u> – John Saunders Oct 24 '11 at 19:34

4 Answers





that is raised when an event from within the user control is fired. This will allow you to bubble the event up the chain so you can handle the event from the form.



When clicking Button1 on the UserControl, i'll fire Button1_Click Which triggers UserControl_ButtonClick on the form:

User control:

```
public event EventHandler ButtonClick;

protected void Button1_Click(object sender, EventArgs e)
{
    //bubble the event up to the parent
    if (this.ButtonClick!= null)
        this.ButtonClick(this, e);
}
```

Form:

```
UserControl1.ButtonClick += new EventHandler(UserControl_E
protected void UserControl_ButtonClick(object sender, Ever
{
    //handle the event
}
```

edited Oct 14 '16 at 22:47



ThunderWiring 476 1 4 21

answered Oct 24 '11 at 19:31



James Johnson 40.8k 6 63 101

3 I can't make the edit, but would you change UserControl1.OnButtonClick to

```
very useful, thaks - Saeid Doroudi Jul 16 '16 at 4:57
```

1 please edit the code of the form: ButtonClick instead of OnButtonClick - ThunderWiring Oct 12 '16 at 14:36

I am having trouble getting this to work on user control instances that are inside repeater control that generates them on the OnltemDatabound event. It is as if the UserControl1.ButtonClick += new EventHandler(UserControl_ButtonClick); part is not doing anything. if (this.ButtonClick!= null) always returns false. — Osprey May 28 '17 at 14:24



Try mapping it. Try placing this code in your UserControl:

18



```
public event EventHandler ValueChanged {
  add { numericUpDown1.ValueChanged += value; }
  remove { numericUpDown1.ValueChanged -= value; }
}
```

then your UserControl will have the ValueChanged event you normally see with the NumericUpDown control.

answered Oct 24 '11 at 19:32



Don't use the *new* keyword here. – Hans Passant Oct 24 '11 at 19:35



you can do like this.....the below example shows text box(user control) value changed

11



```
// Declare a delegate
public delegate void ValueChangedEventHandler(object sende
public partial class SampleUserControl : TextBox
    public SampleUserControl()
        InitializeComponent();
   // Declare an event
    public event ValueChangedEventHandler ValueChanged;
    protected virtual void OnValueChanged(ValueChangedEven
        if (ValueChanged != null)
            ValueChanged(this,e);
   private void SampleUserControl_TextChanged(object send
        TextBox tb = (TextBox)sender;
        int value;
        if (!int.TryParse(tb.Text, out value))
            value = 0;
        // Raise the event
       OnValueChanged( new ValueChangedEventArgs(value));
```

edited Dec 3 '11 at 20:17

answered Oct 24 '11 at 19:44



Home

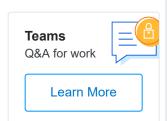
PUBLIC



Tags

Users

Jobs





one of the easy way to do that is use landa function without any problem like





answered Apr 6 '18 at 14:52

