

Re: Null pointer exception for events in usercontrols (C#)

Source:

<http://www.tech-archive.net/Archive/DotNet/microsoft.public.dotnet.languages.csharp/2004-03/0836.html>

From: Stoitcho Goutsev \((100)\) [C# MVP] (100_at_100.com)

Date: 03/03/04

Date: Wed, 3 Mar 2004 11:57:53 -0500

Hi Mark,

Your events are not instantiated until somebody adds an event handler to the event. So the right way to do it, I believe, is to follow the design pattern used throughout Windows Forms framework.

1. In your base class declare the events as you do

```
public event SomethingDoneEventHandler SomethingDone;
```

If you are not familiar with what this line exactly do you can find that info in the c# specs for example.

I believe it is a good practice to follow the name convention suggested by MS

– Give the event some name containing **verb** and not containing the word 'Event'.

e.g. SomethingDone

– Name the delegate after the event + 'EventHandler'

e.g. SomethingDoneEventHandler

– Name the event related data after the event + 'EventArgs'

e.g. SomethingDoneEventArgs

2. Declare one protected virtual OnXXXX method that raises that event (in your base class) named after the event with On prefix.

```
protected virtual OnSomethingDone(SomethingDoneEventArgs e)
{
    if(SomethingDone != null)
        SomethingDone(this, e);
}
```

In this way if you want to catch an event in some inherited class you can simply override the OnXXXX method. Normally you should call the base.OnXXXX method to have the event fired. If you want to block the event don't call

the base implementation.

```
--
HTH
B\rgds
100 [C# MVP]
"Mark" <msaccomandi@libero.it> wrote in message
news:f13e0727.0403030815.56912a75@posting.google.com...
> Hi,
>
> I'm writing user controls with custom events.
>
> I have a parent custom event that exposes some abstract methods and
> some custom events.
>
> I have also created some new user controls that derive from the parent
> custom control and add some text boxes, labels, buttons etc...
>
> So, let's say I created a control with a button in it. I'd want that
> when I click on the button the button sends an event to the custom
> control and the custom control raises a custom event to the form that
> host's it.
>
> Here's an example of my code:
>
> public delegate void PirEventHandler(Object sender, EventArgs e);
>
> public abstract class PirParentCtrl : System.Windows.Forms.UserControl
> {
>     public event PirEventHandler PirEvent;
>     public event PirEventHandler PirClick;
>     public event PirEventHandler PirDoubleClick;
>
>     ...
>
>     protected void ThrowEvent(object sender, System.EventArgs e)
>     {
>         try
>         {
>             PirEvent(this,e);
>         }
>         catch (Exception ex)
>         {
>             MessageBox.Show(this.Name + " (CustomEvent): " + ex.Message);
>         }
>
>         ...
>     }
>
> And this is a custom control:
>
> public class PirButton : PiranhaControls.PirParentCtrl
> {
>
>     private DevExpress.XtraEditors.SimpleButton button1;
>
>     private System.ComponentModel.Container components = null;
>
>     public PirButton()
>     {
>         InitializeComponent();
>     }
> }
```

```
>
>
>     protected override void Dispose( bool disposing )
>     {
>         ...
>     }
>
>     #region Codice generato da Progettazione componenti
>
>
>     private void InitializeComponent()
>     {
>         ...
>     }
>
>     #endregion
>
>     private void button1_Click(object sender, System.EventArgs e)
>     {
>         ThrowEvent(sender, e);
>     }
>
>     public override object GetValue()
>     {
>         return button1.Text;
>     }
>
>     public override void SetValue(params object[] array)
>     {
>         string str=(string)array[0];
>         button1.Text=str;
>     }
> }
>
> When I try to use this control, it works fine when I use its methods
> and properties, but I get a null pointer exception when I click on the
> button. The exception is referred to PirEvent as if it was not
> instantiated. The fact is that I never instantiate this object as I
> thought it is automatically instantiated as soon as the event occurs,
> and I don't know how I could even do it.
>
> What am I doing wrong?
>
> Thanks for your help,
>
> Mark Saccomandi
```