

## Re: I give in...

**Source:**

<http://www.tech-archive.net/Archive/DotNet/microsoft.public.dotnet.framework.aspnet/2004-03/6505.html>

---

**From:** Scott (noemail\_at\_this-is-extra-hotmail.com)

**Date:** 03/19/04

Date: Fri, 19 Mar 2004 08:13:35 -0500

Yes, you can save some of the private members using the ViewState; if your data collection is not that big, you can try to persist in the ViewState (keep in mind that the ViewState can become quite large, and one doesn't want to go crazy adding stuff to it).

Yes, round trips to the server, mean that the control heirarchy will almost always be recreated each time.

I'm not exactly sure what you are doing; but event handlers, like cmdSubmit\_Click are handled early. What I think you are doing in Page\_Load, a data bind to get the radio boxes, should be done in the CreateChildControls (don't just create the RadioButtonList, actually populate it with a datasource/databind).

"nobody" <nobody@no.server> wrote in message news:OAc0n0XDEHA.3664@TK2MSFTNGP10.phx.gbl...  
> Ok, I added the override for the Controls property, and I changed the  
> CreateChildControls override to actually create the RadioButtonList (I was  
> doing "Private m\_RadioButtonList as new WebControls.RadioButtonList"... I've  
> dropped the "New"), fill it in from a private ListItemCollection variable  
> (exposed through a property and filled in via code in the .aspx's Page\_Load  
> event), add it to me.controls, and then set me.ChildControlsCreated = True.  
>  
> I also changed the Answer Get property to call me.EnsureChildcontrols and  
> then Return m\_RadioButtonList.Selectedvalue. When I run the project, the  
> control shows on the web page just fine. All the choices added to  
> qbxQuestion (my control on the page) are there. I select one of the  
> choices, and click the "Submit" button. This fires a cmdSubmit.Click event  
> and the .aspx handles it in cmdSubmit\_Click(...). This just does a  
> lblAnswer.Text = "Answer: " & qbxQuestion.Answer. That's the only line of  
> code in the cmdSubmit\_Click handler. It hits an error. m\_RadioButtonList  
> is Nothing, and m\_Choices (the ListItemCollection that was filled in by the  
> Page\_Load event) is empty. It's as if a new instance of the control was  
> created when the Submit button was clicked. Is this normal? Is every  
> interaction the user does with the server going to create a new instance of  
> all the controls on the page? Is there something I need to do to "persist"  
> the private member variables of my control?  
>  
> "Scott" <noemail@this-is-extra-hotmail.com> wrote in message

> news:uNNQ1DPDEHA.3016@TK2MSFTNGP11.phx.gbl...  
>> Sounds like you are using VB; here's some code in C# (I'm typing this in  
>> OE, so I'm not sure it will even compile). The "trick" here to make sure  
>> that your "inner" (composed?) control gets created in the right spot. The  
>> ideal place is `CreateChildControls` (which, given the name makes some  
>> sense).  
>> The second "trick" is to make sure that `CreateChildControls` is called in  
> the  
>> "right" place when the page controls are being reconstructed, this is done  
>> using the override on `Controls` (the other obvious place to add new  
> controls  
>> is `OnInit` -- given the name -- but that's almost always not the right  
> place  
>> to do it). The last little trick, is to include the `INamingContainer`  
>> interface in the inheritance...  
>>  
>> If you have problems with this, you should turn on page tracing (in your  
>> ASPX file) and add some `Page.Trace.Write` statements; it will really help  
> to  
>> get a handle on where/when things are happening (make sure you wrap the  
> `Page`  
>> access around an `IF` statement to check if `Page` is null -- for example: if  
>> `(this.Page != null) this.Page.Trace.Warn("Hello trace world");` )  
>>

>> There are several good books on this:

>>  
>>  
>>  
>

[http://www.amazon.com/exec/obidos/tg/detail/-/0735615829/qid=1079618092/sr=8-1/ref=sr\\_8\\_xs\\_ap\\_i1\\_xgl14/002-](http://www.amazon.com/exec/obidos/tg/detail/-/0735615829/qid=1079618092/sr=8-1/ref=sr_8_xs_ap_i1_xgl14/002-)

>>

>> And if you are just getting started, this is a must have:

>>  
>>  
>>  
>

[http://www.amazon.com/exec/obidos/tg/detail/-/0735619034/qid=1079618134/sr=8-1/ref=sr\\_8\\_xs\\_ap\\_i1\\_xgl14/002-](http://www.amazon.com/exec/obidos/tg/detail/-/0735619034/qid=1079618134/sr=8-1/ref=sr_8_xs_ap_i1_xgl14/002-)

>>

>>

>> `public class YourCompositeControl : System.Web.UI.Control,`

> `INamingContainer`

>> {

>> `public override ControlCollection Controls`

>> {

>> `get`

>> {

>> `this.EnsureChildControls();`

>> `return base.Controls;`

>> }

>> }

>>

>> `private RadioButtonList m_RadioButtonList;`

>> `protected override void CreateChildControls()`

```
>> {
>> this.Controls.Clear();
>> m_RadioButtonList = new RadioButtonList();
>> // add buttons to your list; could be a data source/data bind
> thing
>> as well
>> m_RadioButtonList.SelectedIndexChanged += new
>> EventHandler(m_RadioButtonList_SelectedIndexChanged);
>> Controls.Add(m_RadioButtonList);
>> this.ChildControlsCreated = true;
>> }
>>
>> string m_answer = string.Empty;
>> public string Answer
>> {
>> get
>> {
>> this.EnsureChildControls();
>> // now if you wanted to be clever you could take a look at the
>> m_RadioButtonList here
>> // and see if there's a selected index and that would keep you
>> from doing this selected index changed
>> // thing --
>> return m_answer;
>> }
>> }
>>
>> private void m_RadioButtonList_SelectedIndexChanged(object sender,
>> EventArgs e)
>> {
>> RadioButtonList rbl = sender as RadioButtonList;
>> if (rbl != null && rbl.SelectedIndex != -1)
>> {
>> m_answer = rbl.SelectedValue;
>> }
>> }
>> }
>>
>>
>>
>> "nobody" <nobody@no.server> wrote in message
>> news:%23jKUmJJDEHA.1228@TK2MSFTNGP11.phx.gbl...
>>> Ok, I give in (not up yet... just in). Can someone *PLEASE* point me to
> a
>>> good "how to" or tutorial on designing composite WebControls? I
> obviously
>>> have no clue what I'm doing. :( I think my problem might be that I'm a
>>> "windows forms" person and this ASP thing is a different creature when
> it
>>> comes to state and when/how you must persist it.
>>>
```

> > > *I'm adding a RadioButtonList to my WebControl*  
> > > *(me.controls.add(m\_RadioButtonList). I have an event handler for the*  
> > > *SelectedIndexChanged to store the currently selected index in m\_Answer*  
> > *and*  
> > *I*  
> > > *have a readonly property Answer to allow the aspx to get the answer from*  
> > *my*  
> > > *control at any time (like when the user clicks a submit button). On the*  
> > > *aspx page, I have a button that just sets a label on the page to my*  
> > > *control's .Answer property. It's blank, even when I've made a selection*  
> > *in*  
> > > *the RadioButtonList. My SelectedIndexChanged event handler is never*  
> > *being*  
> > > *called to set m\_Answer. And also, when the button is clicked, it*  
> > *redraws*  
> > > *the page (with the blank "Answer" label) with all the options in the*  
> > > *RadioButtonList in my control gone.*  
> > >  
> > >  
> >  
> >  
> >  
>  
>