How to get readonly textbox value in codebehind file

Published 6/13/2013 by Raghav Khunger

In this article I will explain how to get readonly textbox value in codebehind file. I am writing this article to resolve these issues:

- ASP.NET Ready Only TextBox lose client side changes, values across post back?
- Why Readonly Text box values are empty in code behind ?
- Issue in Retrieving textbox value when readonly = true in codebehind.?

I will discuss the case when we have asp.net textbox control with readonly attribute set to true, and we have changed the value in textbox with javascript. Now the issue comes when we want to retrieve that changed value of that textbox in codebehind. As a security measure in the asp.net the readonly fields and disabled fields lose their values in post backs.... ASP.NET 2.0 had a design change by which the <asp:TextBox> control marked with its ReadOnly property as true, would ignore client side changes and would lose the same across postback. ASP.Net does not accept changes made to the text of a ReadOnly TextBox sent from the client, the LoadPostData gets short circuited before the 'Text' property on the TextBox is reset using the post data in this case. The change was made to ensure that 'ReadOnly' textboxes really behave as readonly. So if we try to modify the text box value or add a value to the text box with javascript we will not be able to retrieve the value in the code behind or simply the value will be lost across postback . This behaviour has been designed with the idea that a ReadOnly TextBox shouldnt be modified in the client side by a malicious code.Readonly is acting as disabled so browsers don't post values back in disabled input controls.

From MSDN:

http://msdn.microsoft.com/en-us/library/system.web.ui.webcontrols.textbox.readonly.aspx

> Use the ReadOnly property to specify whether the contents of the TextBox control can be changed. Setting this property to true will prevent users from entering a value or changing the existing value. Note that the user of the TextBox control cannot change this property; only the developer can.The Text value of a TextBox control with

> the ReadOnly property set to true is sent to the server when a postback occurs, but the server does no processing for a read-only text box. This prevents a malicious user from changing a Text value that is read-only. The value of the Text property is preserved in the view state between postbacks unless modified by server-side code. This property cannot be set by themes or style sheet themes.

Microsoft's reply to a bug report related to topic.

http://connect.microsoft.com/VisualStudio/feedback/ViewFeedback.aspx?FeedbackID=102065

> If you have a control on a page that is marked Read-Only and EnableViewState is set to false on the Page, the ReadOnly value will no longer post back in ASP.NET 2.0 - the value gets lost. This even though the value is actually returned in the POST buffer.
>
> This behavior is also different than 1.1 which (correctly I say) posted back the READONLY value.

Readonly textbox value is not sent back during postback , the page can remember these values if they are specified directly in the markup portion of the ASP.NET page in the textbox's control's declarative syntax or if they are set programmatically and ViewState is enabled for the control and page. If ViewState is disabled and the control value is specified programmatically, it will be lost on postback because the actual value in the disabled TextBox won't be sent back to the server during postback.

Workaround this issue
Instead of making the textbox readonly using properties option. Add the code:

TextBox1.Attributes.Add("readonly", "readonly");

In codebehind which will still make the textbox as readonly and will also retain the value after each postback.

Consider the following test code to clear that.

Aspx Code

```
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
<title></title>

<script type="text/javascript" >

function addText1()
{
var textBox1 = document.getElementById('<%= TextBox1.ClientID %>');
textBox1.value = 'Test Text1';
}

function addText2()
{
var textBox2 = document.getElementById('<%= TextBox2.ClientID %>');
textBox2.value = 'Test Text2';
}

</script>
</head>
<body>
<form id="form1" runat="server">
<div>
Textbox set as readonly in .aspx
<asp:TextBox ID="TextBox1" runat="server" ReadOnly="true" ></asp:TextBox><input
id="btnAddText1" type="button" value="Add text in this textbox" onclick="addText1();" />
<br />

Textbox set as readonly in .aspx.cs ( codebehind)
<asp:TextBox ID="TextBox2" runat="server" ></asp:TextBox><input
id="btnAddText2" type="button" value="Add text in this textbox" onclick="addText2();" />
<br />

<asp:Button ID="btnSubmit" runat="server" Text="Submit"
onclick="btnSubmit_Click" /> <br />

After postback value in Textbox set as readonly in .aspx  :<asp:Label ID="Label1" runat="server" Text=""></asp:Label><br />
After postback value in Textbox set as readonly in .aspx.cs ( codebehind)   :<asp:Label ID="Label2" runat="server" Text=""></asp:Label><br />


</div>
</form>
</body>
</html>
```

Codebehind:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

public partial class how_to_get_readonly_textbox_value_in_codebehind : System.Web.UI.Page
{
protected void Page_Load(object sender, EventArgs e)
{
if(!IsPostBack)
{
TextBox2.Attributes.Add("readonly","readonly");
}
}
protected void btnSubmit_Click(object sender, EventArgs e)
{
Label1.Text = TextBox1.Text;
Label2.Text = TextBox2.Text;
}
}
```

In the above example I have used two textboxes one is marked as readonly in the markup in aspx
<asp:TextBox ID="TextBox1" runat="server" ReadOnly="true" ></asp:TextBox>

Second textbox is set as readonly true in codebehind ie
In aspx:
<asp:TextBox ID="TextBox2" runat="server" ></asp:TextBox>
In Codebehind:
TextBox2.Attributes.Add("readonly","readonly");
There are two html buttons in front of each textbox on click of each button text is inserted in corresponding textbox through javascript.Also there are two labels whose values are set in code behind with the values of corresponding textboxes in postback so as to display the values of textbox.There is one Asp.net submit button so as to make a postback.
After running the above code and doing postback with asp.net submit button you will see that first label value is blank because as explained above textbox1 text is set as readonly in markup and second label value contains the value of textbox2 whose readonly attribute is set in code behind. So what magic happens writing that in codebehind :
What happens behind the scenes is that the client sends along the value of the textbox set as readonly through the form values, but the ASP.NET 2.0 engine does not take that value and assign it to the Text property of the textbox on postback to help protect against a malicious user changing the read-only textbox value themselves.

Another good explanation I found by Andrew Hare at
Why does a read-only textbox does not return any data in ASP.NET?

> There is a little bit of strangeness when it comes to the ASP.NET Readonly property and the readonly attribute of an HTML input element. Rather than setting the Readonly property of the web control try simply adding the HTML attribute to the control like this:
> textBox.Attributes.Add("readonly", "readonly");
> This will make the control read-only in the client's browser yet still allow you to retrieve the value of the input when it posts back to the server.

Do let me know your feedback,comments.

tags : Readonly textbox