

The results are in! See what nearly 90,000 developers picked as their most loved, dreaded, and desired coding languages and more in the 2019 Developer Survey.

groupname doesn't work in more than one radiobutton inside repeater asp.net

[Ask Question](#)


9



1

I have a repeater and inside the repeater a radiobutton control, in code behind I fill the groupname for the radiobutton control, so, when I run it, I have a table with many rows and some of them have radiobutton:

```
<asp:updatepanel id="UpdatePanel1" runat="server" updatemode="Conditional">
  <ContentTemplate>
    <asp:Repeater ID="Repeater1" runat="server" ViewStateMode="Enabled">
      <HeaderTemplate>
        <table class="table table-responsive table-bordered ">
          <tr class="text-center" style="background-color: #6e6259; color:
white;">
            <th class="text-center">DESCRIPTION</th>
          </tr>
        </table>
      </HeaderTemplate>
      <ItemTemplate>
        <tr>
          <td style="padding-left: 20px;">
            <asp:RadioButton ID="rbtDinamic" OnCheckedChanged="rbtDinamic_CheckedChanged"
AutoPostBack="true"
              ViewStateMode="Enabled" Visible="false" GroupName='<%=Eval("groupvalue") %>'
runat="server"/></td>
          </tr>
        </ItemTemplate>
      <FooterTemplate>
        <table>
          <tr>
            <td></td>
          </tr>
        </table>
      </FooterTemplate>
    </asp:Repeater>
  </ContentTemplate>
</asp:updatepanel>
```

And in the itemdatabound of repeater I fill the value for groupname:

```
Private Sub Repeater1_ItemDataBound(sender As Object, e As RepeaterItemEventArgs)
Handles Repeater1.ItemDataBound
    Try
        If e.Item.ItemType = ListItemType.AlternatingItem Or e.Item.ItemType =
ListItemType.Item Then
            If CType(e.Item.FindControl("hdf1"), Label).Text = False Then
                CType(e.Item.FindControl("rbtDinamic"), RadioButton).Visible = True
                CType(e.Item.FindControl("rbtDinamic"), RadioButton).GroupName =
CType(e.Item.FindControl("groupvalue"), Label).Text = False
            End If
        End If
    Catch ex As Exception
    End Try
End Sub
```

But when I run it the repeater creates the group name with diferent names:

```
Radiobutton row 1:
Repeater1$ctl05$1
```

```
Radiobutton row 2:

Repeater1$ctl06$1
```

So it let checked all the radiobuttons, instead to uncheck when another one for the same group is cheked.

I find this code in a forum, but it work only if I have only one groupname, but I can have more than one groupname:

```
Protected Sub rbtDinamic_CheckedChanged(sender As Object, e As EventArgs)
    For Each item As RepeaterItem In Repeater1.Items
        Dim rbtn As RadioButton = DirectCast(item.FindControl("rbtDinamic"),
RadioButton)
        rbtn.Checked = False
    Next
    DirectCast(sender, RadioButton).Checked = True
End Sub
```

But there can be more than one group of radiobuttons, so in this case I can't use this code.

Is there anywhere to do this? thanks

c# asp.net vb.net radio-button repeater

edited Sep 14 '17 at 11:50

asked Sep 14 '17 at 11:41



Esraa_92

809 1 9 32

3 Answers



8



This is a known bug related with `RadioButton` control usage inside `ItemTemplate` OR `AlternatingItemTemplate` ([more info](#)). This caused by `Repeater` mangling the naming of control ID & group names which assigned automatically in background (assumed using dynamic `ClientIDMode`). To fix this issue, set up a client-side function like this:



+50

```
function setExclusiveRadioButton(name, current)
{
    regex = new RegExp(name);

    for (i = 0; i < document.forms[0].elements.length; i++)
    {
        var elem = document.forms[0].elements[i];
        if (elem.type == 'radio')
        {
            elem.checked = false;
        }
    }
}
```

Home

PUBLIC

Stack Overflow

Tags

Users

Jobs

Teams

Q&A for work



[Learn More](#)

```

    }
    current.checked = true;
}

```

Later, set the script targeting the radio button control as given below:

```

Private Sub Repeater1_ItemDataBound(sender As Object, e As Repeater1
Handles Repeater1.ItemDataBound
    Try
        If e.Item.ItemType = ListItemType.AlternatingItem Or e.Item.
ListItemType.Item Then
            If CType(e.Item.FindControl("hdf1"), Label).Text = False
                CType(e.Item.FindControl("rbtnDinamic"), RadioButton)
                CType(e.Item.FindControl("rbtnDinamic"), RadioButton)
                CType(e.Item.FindControl("groupvalue"), Label).Text = False
            End If
        End If

        ' put the proper client-side handler for RadioButton
        Dim radio As RadioButton = CType(e.Item.FindControl("rbtnDina
        Dim script As String = "setExclusiveRadioButton('Repeater1.*
[RadioButton_GroupName]', this)"

        radio.Attributes.Add("onclick", script)

    Catch ex As Exception
    End Try
End Sub

```

NB: The first argument of `setExclusiveRadioButton` method should be set to this regex convention: `[repeater control ID].*[RadioButton_GroupName]` (`RadioButton_GroupName` value may be retrieved using `Eval`). Alternatively you can use basic HTML `input type="radio"` or use `RadioButtonList` instead.

Reference:

[Using RadioButton Controls in a Repeater](#)

Similar issues:

[radiobutton inside repeater](#)

[only one radiobutton selection in repeater](#)[ASP.NET - Radio Buttons In Repeaters](#)

answered Sep 20 '17 at 10:09



Tetsuya Yamamoto

17.1k 4 23 42



On the client side, set the `name` of the radio to whatever group you like, BUT write down the generated name in a `data-` attribute.

2



Then, just before form submission, copy the `data-` attribute back to the `name` attribute, so the ASP.NET could recognize the control on the server.

This script will do it:

```
<script type="text/javascript">
    $(document).ready(function (e) {
        $("input[type='radio']").each(function (idx, elm) {
            var generatedName = $(elm).attr("name");
            $(elm).data("name", generatedName);
            $(elm).attr("name", "whatever-group-name");
        });
    });

    function onSubmit() {
        $("input[type='radio']").each(function (idx, elm) {
            var generatedName = $(elm).data("name");
            $(elm).attr("name", generatedName);
        });
    }
</script>
```

To detect the form submission, you do call

RegisterOnSubmitStatement . E.g. in your Page_Load :

```

if (!IsPostBack)
{
    Page.ClientScript.RegisterOnSubmitStatement(Page.GetType(), "pre|
onSubmit();");
}

```

edited Sep 28 '17 at 6:10

answered Sep 26 '17 at 9:42



Bozhidar Stoyneff

2,985 1 6 21

thanks for your answer, I have a geusetion, what is the value for idx and elm? – [Esraa_92](#) Oct 16 '17 at 15:23

They contain the *index of the element in the array* and the *element* itself, respectively. jQuery will provide them for you; no need you to do anything to initialize them. Just use them inside the body of the function, passed to the `.each` method. – [Bozhidar Stoyneff](#) Oct 17 '17 at 0:34



5



As other user provided the root cause of the problem , so i wouldn't be explaining same but i would provide you with the **Jquery** based solution:

```

jQuery("[name$='optValue']").attr("name",jQuery("[name$='optValue'
jQuery("[name$='optValue']").click(function (){
    //set name for all to name of clicked
    jQuery("[name$='optValue']").attr("name", this.attr(
    });

```

with `attr("name",jQuery("[name$='optValue']")` will try to select all of the inputs on the page which ending with `optValue` i.e. `optValue`

items in the repeater. after that it changes the `name` attribute to first value found for all the `optValue` elements. The `attr("name")` function (used here in 'get' format) always returns the first in the list. So all the option buttons get the same 'name' attribute in the Html, which allows the select to work correctly.

A great Work-around from [this Source](#)

answered Sep 20 '17 at 11:11



Webruster

7,505 4 24 40