

# Manipulating RadGrid on server side

There are some real-life requirements when we need to hide some columns and/or conditionally form 'em or have to add links to the Gridview. This article describes how we can:

- 1) **Hide columns** dynamically
- 2) **Add dynamic controls** (linkbuton etc) some provision for *conditional formatting* with client-side scripting and can make the things really simple and also how can we hide columns from the grid that we need in the application; (E.g. the PK or FK etc.)

A bit custom, not very popular but interesting scenario that may add aesthetics to your web-application.

## Hiding Columns

A column can be hidden from Gridview in a scenario where we need to query a particular column from the database, but not needed to be displayed in the grid itself.

We can leverage the “**OnItemDataBound**” event and can change properties after casting **e.Item** to **GridDataItem**

---

```
protected void RadGrid1_ItemDataBound(object sender, GridItemEventArgs e)
{
    GridDataItem thisGridDatatem = e.Item as GridDataItem;
    if (thisGridDatatem["column_name"] != null)
        thisGridDatatem["id"].Visible = false;
}
```

---

---

---

Of course in your application you *may* want to do this conditionally (role based and permission/access-rights bases scenarios). Checking a column against null is a good practice. Sure, we can take it to the next level in the case when we just need to hide a column we may do something like following:

---

```
GridDataItem thisGridDatatem = e.Item as GridDataItem;
if ( e.Item.ItemType != GridItemType.Header )
{
    if (thisGridDatatem["column_1"] != null)
        thisGridDatatem["column_1"].Visible = false;
    else if (thisGridDatatem["column_2"] != null)
        thisGridDatatem["column_2"].Visible = false;
}
```

---

### **Adding (client-side ready) Controls Dynamically**

This recipe is a best resource in the cases when users want to ‘click’ on a text and want to drill down the result. One possible approach here would be to add links to the cell text and ***either***, to query the database with appropriate parameters from the server-side ***or*** to call your JavaScript function for Ajax-Call from client-side. Most commonly these parameters are the ‘text’ of adjacent cells in the same row.

Either way, we’d be needing to know what we need to use for parameters, here is a possible approach regardless the analogy being followed. Let’s go back to OnItemDataBound event:

---

```
protected void RadGrid1_ItemDataBound(object sender, GridItemEventArgs e)
```

```
{
    if (e.Item is GridDataItem && thisGridDatatem != null)
    {
        if (thisGridDatatem["target_column"] != null)
        {
            thisGridDatatem["id"].Controls.Clear();

            LinkButton lb = new LinkButton();

            lb.Text = ((System.Data.DataRowView)(e.Item.DataItem)).Row[3].ToString();

            lb.Attributes.Add("attrib1", ((System.Data.DataRowView)(e.Item.DataItem)).Row[2].ToString());
            // Some Real Life Attribute

            lb.Attributes.Add("attrib2", ((System.Data.DataRowView)(e.Item.DataItem)).Row[4].ToString());
            // Some Real Life Attribute

            lb.Attributes.Add("attrib3", ((System.Data.DataRowView)(e.Item.DataItem)).Row[5].ToString());
            // Some Real Life Attribute

            lb.Command += new CommandEventHandler(lb_Command);

            lb.ID = "someConvenientName" + e.Item.RowIndex.ToString();

            TableCell c1 = new TableCell();

            c1.Controls.Add(lb);

            e.Item.Cells.RemoveAt(1);

            e.Item.Cells.AddAt(1, c1);
        }
    }
}
```

---

```
void lb_Command(object sender, CommandEventArgs e)
{
    LinkButton lb = (LinkButton)sender;
    string
        casenum = lb.Text
        , attrib1 = (lb.Attributes["attrib1"] ?? "").ToString()
        , attrib2 = (lb.Attributes["attrib2"] ?? "").ToString()
        , attrib3 = (lb.Attributes["attrib3"] ?? "").ToString();
}
```

---

Now if you go back to your browser and look at the rendered code you may find something as following, attributes are also available there. Now you may refer to this control by using selectors using javascript or javascript library like jQuery

Product <span>×</span>			
Product	Unit_Price	Disc	Extended_Price
Product: 1			
5	18.0000	20	72
45	18.0000	0	810
80	18.0000	20	1152
Product: 2			
100	19.0000	25	1425
3	19.0000	0	57
30	19.0000	25	427.5
25	15.2000	25	285
45	15.2000	15	581.4
Product: 5			

File Find Disable View Images Cache Tools Validate Browser Mode: IE10 Document Mode: Standards

HTML CSS Console Script Profiler Network



```
lass="rgRow" id="RadGrid1_ctl00__0">
```

```
t - Empty Text Node
```

```
class="rgGroupCol">
```

```
>
```

```
<a href="javascript:__doPostBack('RadGrid1$ctl00$ctl05$ctl00','')>
```

```
>
```

```
>
```

```
>
```

```
t - Empty Text Node
```

Style Trace Styles Layout Attributes

Attribute: Node: a

Name

attrib3

attrib2

attrib1

href

Value

72

20

Chai

javascript:\_\_doP