

## ASP.NET Grid – How to Delete Selected Rows

With the ASPxGridView, it is possible to enable and use the built-in links to delete each row. While this is a very useful feature, there may be cases where you would want your users to select and delete multiple rows at once by clicking a button.

In this video, you will learn how to add code to delete selected grid rows at runtime.

So let's get started...

1. I'll start with an ASP.NET Application that has an ASPxGridView control bound to the Departments table of the "Departments" Access Database.
2. Let's take a look at the ASP.NET code.
3. You can see that I have also specified a **DeleteCommand** for the Access Data Source.
4. Back in the designer, I'll open the grid's smart tag and enable the Selection Checkbox so that multiple rows can be selected at runtime.
5. To access the grid from other controls, I'm going to set its **ClientInstanceName** property.
6. I'll call it "dGrid".
7. Next, I drop an ASPxButton onto the form.
8. I set its **Text** property to "Delete Selected Rows".
9. Now, by default, the button will perform a postback when clicked.
10. Since we don't want the page to be completely refreshed, I'm going to set the **AutoPostBack** property of the ASPxButton control to *False*.
11. Now I'll handle the button's client-side click event.
12. I open its smart tag and click on the "Client-Side Events" link.

13. This brings up the Client-Side Events Editor.

14. I select the **Click** event and add the following code to the Event Handler Body.

```
function(s, e) {  
    dGrid.PerformCallback();  
}
```

15. The **PerformCallback** method is called to send a callback to the server and raise the **CustomCallback** event of the ASPxGridView.

16. To perform the deletion, I need to handle the CustomCallback event.

17. I select the grid control and create the event handler.

18. Here, I create a List and populate it with the key values of the selected rows.

19. For this, I call the **GetSelectedFieldValues** method of the ASPxGridView.

20. Then I delete the selected records.

21. Finally the **DataBind** method is called to refresh the grid.

22. And that's it!

23. I run the web application.

24. I'll select a few rows...

25. ... and click on the "Delete Selected Rows" button.

26. The rows are deleted from the data source and the grid is automatically refreshed.

The CustomCallback event can be used to perform any type of server-side processing. For more information, please refer to the ASPxGridView Documentation

(<http://www.devexpress.com/Help/Content.aspx?help=ASPxGridView>).

Thanks for watching and thank you for choosing DevExpress!

## Code [ASP.NET]

```
<dxwgv:ASPGridView ID="ASPGridView1" runat="server"
    AutoGenerateColumns="False" ClientInstanceName="dGrid"
    DataSourceID="AccessDataSource1" KeyFieldName="ID"
    oncustomcallback="ASPGridView1_CustomCallback" Width="411px">
    <Columns>
        <dxwgv:GridViewCommandColumn ShowSelectCheckbox="True" VisibleIndex="0">
        </dxwgv:GridViewCommandColumn>
        <dxwgv:GridViewDataTextColumn FieldName="ID"
            ReadOnly="True" VisibleIndex="1">
        </dxwgv:GridViewDataTextColumn>
        <dxwgv:GridViewDataTextColumn FieldName="PARENTID" VisibleIndex="2">
        </dxwgv:GridViewDataTextColumn>
        <dxwgv:GridViewDataTextColumn FieldName="IMAGEINDEX" VisibleIndex="3">
        </dxwgv:GridViewDataTextColumn>
        <dxwgv:GridViewDataTextColumn FieldName="DEPARTMENT" VisibleIndex="4">
        </dxwgv:GridViewDataTextColumn>
        <dxwgv:GridViewDataTextColumn FieldName="BUDGET" VisibleIndex="5">
        </dxwgv:GridViewDataTextColumn>
        <dxwgv:GridViewDataTextColumn FieldName="LOCATION" VisibleIndex="6">
        </dxwgv:GridViewDataTextColumn>
        <dxwgv:GridViewDataTextColumn FieldName="PHONE1" VisibleIndex="7">
        </dxwgv:GridViewDataTextColumn>
        <dxwgv:GridViewDataTextColumn FieldName="PHONE2" VisibleIndex="8">
        </dxwgv:GridViewDataTextColumn>
    </Columns>
</dxwgv:ASPGridView>
<asp:AccessDataSource ID="AccessDataSource1" runat="server"
    DataFile="~/App_Data/Departments.mdb"
    SelectCommand="SELECT [ID], [PARENTID], [IMAGEINDEX],
        [DEPARTMENT], [BUDGET], [LOCATION], [PHONE1], [PHONE2]
        FROM [Departments]"
    DeleteCommand="DELETE FROM [Departments] WHERE [ID] = ?">
    <DeleteParameters>
        <asp:Parameter Name="ID" Type="Int32" />
    </DeleteParameters>
</asp:AccessDataSource>
<dx:ASPButton ID="ASPButton1" runat="server"
    Text="ASPButton" Width="177px" AutoPostBack="False">
    <ClientSideEvents Click="function(s, e) {
        dGrid.PerformCallback();
    }" />
</dx:ASPButton>
```

## Code [C#]

```
protected void ASPxGridView1_CustomCallback(object sender,
DevExpress.Web.ASPxGridView.ASPxGridViewCustomCallbackEventArgs e)
{
    // Obtains the key values of selected rows.
    List<object> keyValues = ASPxGridView1.GetSelectedFieldValues("ID");
    StringBuilder sb = new StringBuilder(keyValues[0].ToString());
    foreach (object key in keyValues)
    {
        AccessDataSource1.DeleteParameters["ID"] = new Parameter("ID",
                                                                    TypeCode.Int32, key.ToString());

        AccessDataSource1.Delete();
    }
    // Refreshes the grid.
    ASPxGridView1.DataBind();
}
```