

Dynamically Adding An Accordion Pane

Christian Wenz

Overview

The Accordion control in the AJAX Control Toolkit provides multiple panes and allows the user to display one of them at a time. Panels are usually declared within the page itself, but server-side code can be used to achieve the same result.

Steps

The Accordion control exposes all important properties to server-side code. Among other things, the **Panes** property grants access to the collection of panes that make up the Accordion. Every pane there is of type **AccordionPane**. It is therefore trivial to create such a pane:

```
AccordionPane ap1 = new AccordionPane();
```

The **HeaderContainer** property of **AccordionPane** provides access to the ASP.NET controls within the header section of the pane; the **ContentContainer** property of **AccordionPane** does the same for the content section of the pane. This allows ASP.NET code to add content to the panes:

```
ap1.HeaderContainer.Controls.Add(new LiteralControl("Using  
Code"));  
ap1.ContentContainer.Controls.Add(new LiteralControl("Adding panes  
using code is really flexible."));
```

Finally, the pane(s) must be added to the **Panes** collection of the Accordion:

```
acc1.Panes.Add(ap1);
```

Here is a complete server-side code that adds two panes to an Accordion control:

```
<script runat="server">  
void Page_Load()  
{  
    if (!Page.IsPostBack)  
    {  
        AccordionPane ap1 = new AccordionPane();  
        ap1.HeaderContainer.Controls.Add(new LiteralControl("Using  
Markup"));  
        ap1.ContentContainer.Controls.Add(new LiteralControl("Adding  
panes using markup is really simple."));  
        AccordionPane ap2 = new AccordionPane();  
        ap2.HeaderContainer.Controls.Add(new LiteralControl("Using  
Code"));  
        ap2.ContentContainer.Controls.Add(new LiteralControl("Adding  
panes using code is really flexible."));  
    }  
}
```

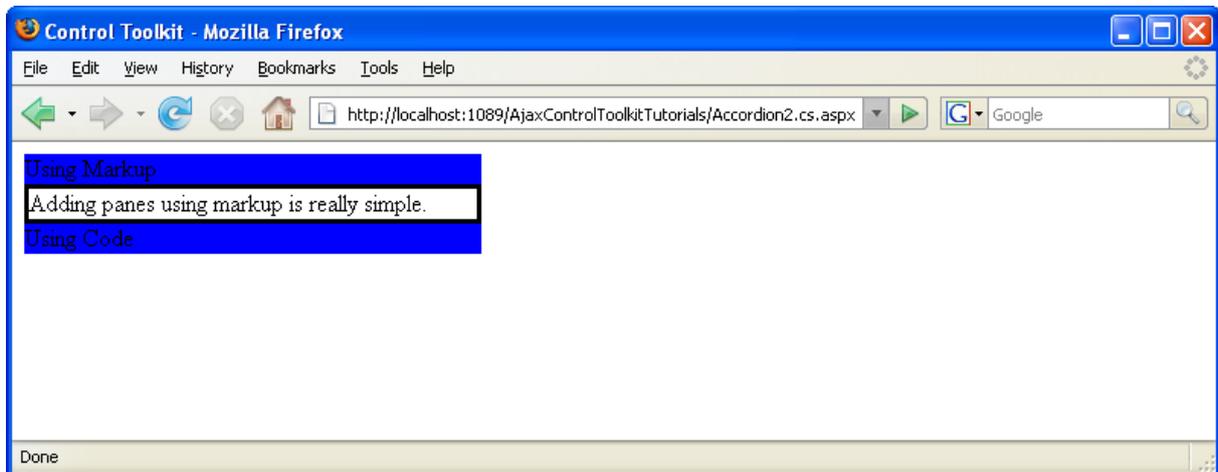
```
    ac1.Panes.Add(ap1);
    ac1.Panes.Add(ap2);
}
}
</script>
```

The only missing element is the Accordion itself, which depends on the presence of the ASP.NET **ScriptManager** control:

```
<form id="form1" runat="server">
  <asp:ScriptManager ID="asm" runat="server" />
  <div>
    <ajaxToolkit:Accordion ID="ac1" runat="server"
      HeaderCssClass="header" ContentCssClass="content"
      Width="300px" FadeTransitions="true">
    </ajaxToolkit:Accordion>
  </div>
</form>
```

To finish the example, the two CSS classes referenced in the Accordion control provide style information for the browser:

```
<style type="text/css">
  .header {background-color: blue;}
  .content {border: solid;}
</style>
```



The data in the accordion was dynamically added by server-side code