Handling Postbacks from A Popup Control With an UpdatePanel

Christian Wenz

Overview

The PopupControl extender in the AJAX Control Toolkit offers an easy way to trigger a popup when any other control is activated. Special care has to be taken when a postback occurs within such a popup.

Steps

When using a **PopupControl** with a postback, an **UpdatePanel** can prevent the page refresh caused by the postback. The following markup defines a couple of important elements:

- A **ScriptManager** control so that the ASP.NET AJAX Control Toolkit works
- Two **TextBox** controls which will both trigger a popup
- A **Panel** control that will serve as the popup
- Within the panel, a **Calendar** control is embedded within an **UpdatePanel** control
- Two **PopupControlExtender** controls that assign the panel to the text boxes

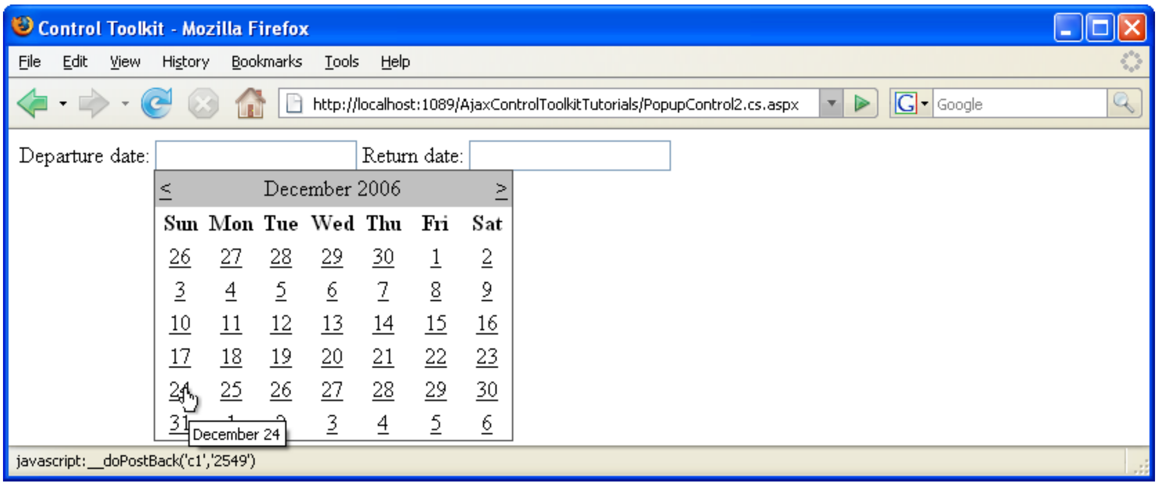
```
<form id="form1" runat="server">
  <asp:ScriptManager ID="asm" runat="server" />
  <div>
    Departure date: <asp:TextBox ID="tbDeparture" runat="server"
   />
    Return date: <asp:TextBox ID="tbReturn" runat="server" />
  </div>
  <asp:Panel ID="pnlCalendar" runat="server">
    <asp:UpdatePanel ID="up1" runat="server">
      <ContentTemplate>
        <asp:Calendar ID="c1" runat="server"
   OnSelectionChanged="c1_SelectionChanged" />
      </ContentTemplate>
    </asp:UpdatePanel>
  </asp:Panel>

  <ajaxToolkit:PopupControlExtender ID="pce1" runat="server"
    TargetControlID="tbDeparture" PopupControlID="pnlCalendar"
   Position="Bottom" />
  <ajaxToolkit:PopupControlExtender ID="pce2" runat="server"
    TargetControlID="tbReturn" PopupControlID="pnlCalendar"
   Position="Bottom" />
</form>
```

Note that the **OnSelectionChanged** attribute of the **Calendar** control is set. So when the user selects a date within the calendar, a postback occurs and the server-side

method **c1_SelectionChanged()** is executed. Within that method, the current date must be retrieved and written back to the textbox.
The syntax for that is as follows: First of all, a proxy object for the **PopupControlExtender** on the page must be generated. The ASP.NET AJAX Control Toolkit offers the **GetProxyForCurrentPopup()** method. The object this method returns supports the **Commit()** method which sends a value back to the control that triggered the popup (not the control that triggered the method call!). The following code provides the selected date as the argument for the **Commit()** method, causing the code to write the selected date back to the text box:

```
<script runat="server">
  protected void c1_SelectionChanged(object sender, EventArgs e)
  {
    PopupControlExtender pce =
   AjaxControlToolkit.PopupControlExtender.GetProxyForCurrentPopup
   (Page);
    pce.Commit((sender as
   Calendar).SelectedDate.ToShortDateString());
  }
</script>
```

Now whenever you click on a calendar date, the selected date appears in the associated text box, creating a date picker control that can currently be found on many websites.


Control Toolkit - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://localhost:1089/AjaxControlToolkitTutorials/PopupControl2.cs.aspx
Google

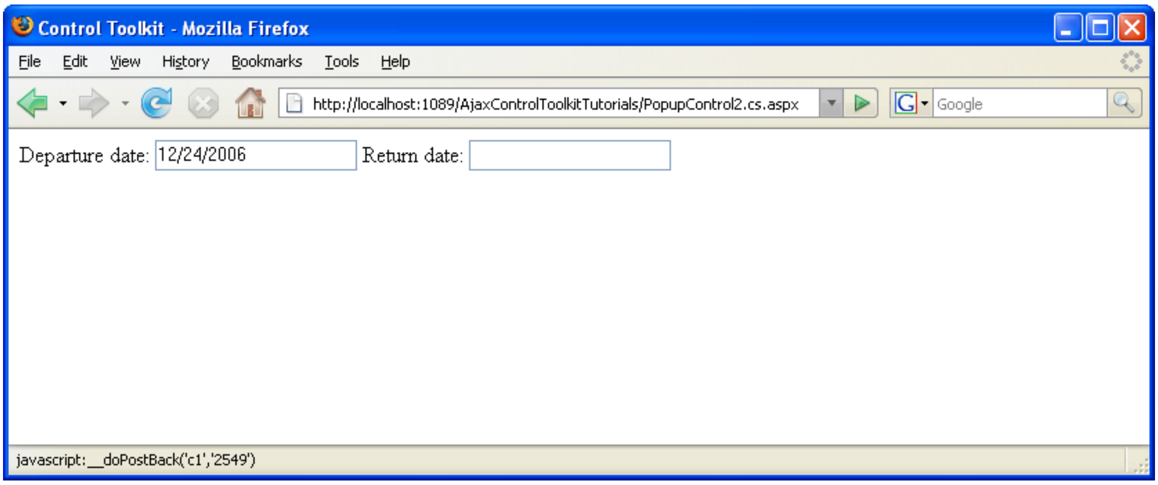
Departure date: Return date:

<			December 2006			>
Sun	Mon	Tue	Wed	Thu	Fri	Sat
26	27	28	29	30	1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30
31	1	2	3	4	5	6

December 24

javascript:__doPostBack('c1','2549')

The Calendar appears when the user clicks into the textbox


Control Toolkit - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://localhost:1089/AjaxControlToolkitTutorials/PopupControl2.cs.aspx
Google

Departure date: 12/24/2006 Return date:

javascript:__doPostBack('c1','2549')

Clicking on a date puts it in the textbox