

Contents

Repeater Control	1
Displaying data with the repeater control	1
Using templates with the repeater control	4
Handling repeater control events	5
References	7

The repeater control is entirely template driven. You can format the rendered output of the control in any way that you please. The repeater control provides you with maximum flexibility in rendering a set of database records.

Displaying data with the repeater control

To display data with the repeater control you must create an item template.

Activity 1. Using the repeater control

default.aspx

In this exercise we will display all the titles from the titles table in the PUBS database as a coma delimited list.

Add a data source control and set it up so that it retrieves the title from the titles table.

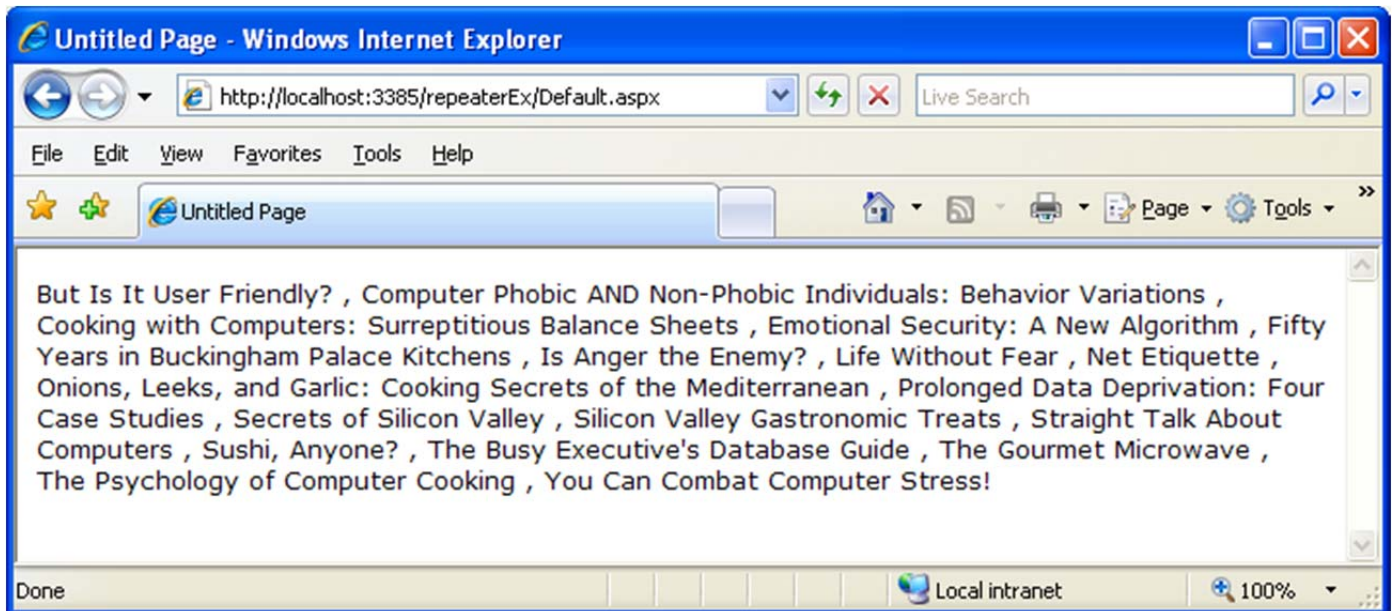
Add a repeater control and allocate the data source control to it. You will see that no templates are available for you to use and nothing is set up for you. If you were going to view the page now there would be no output.

You need to switch over to source view and add the template in yourself. Add the following code to you file in source view:

```
<div>
<asp:SqlDataSource ID="sdsTitles" runat="server"
ConnectionString="<%$ ConnectionStrings:ConnectionStringPubs %>"
SelectCommand="SELECT [title] FROM [titles]">
</asp:SqlDataSource>

<asp:Repeater ID="Repeater1" runat="server"
DataSourceID="sdsTitles">
<ItemTemplate>
    <%# Eval("Title") %>
</ItemTemplate>
<SeparatorTemplate>
    ,
</SeparatorTemplate>
</asp:Repeater>
</div>
```

So you can see with the repeater control you have total control on how it's going to be displayed but the control does nothing for you other than repeat the data.



Activity 2. Repeater and div

titles.aspx

Using a div tag in the repeater can specify how each row of data will be displayed. This div can be formatted using a style sheet.

Create a web page with a datasource control set up the data source control so that it retrieves the title, price date published and notes from the titles table.

Add a repeater control and set it up in source view:

```
<asp:SqlDataSource ID="sdsTitles" runat="server"
ConnectionString="<%$ ConnectionStrings:ConnectionStringPubs %%"
SelectCommand="SELECT [title], [price], [pubdate], [notes]
FROM [titles]">
</asp:SqlDataSource>

<asp:Repeater ID="Repeater1" runat="server" DataSourceID="sdsTitles">
    <ItemTemplate>
        <div class="book">
            <asp:Label ID="lblTitle" runat="server"
Text='<%# Eval("title") %>'
CssClass="bold"></asp:Label>
            <br />
            <asp:Label ID="lblPrice" runat="server"
Text='<%# Eval("price", "{0:C}") %>' >
</asp:Label>
            Published: <asp:Label ID="lblDate" runat="server"
Text='<%# Eval("pubDate", "{0:d}") %>' ></asp:Label>
            <br />
            <div class="notes">
                <asp:Label ID="lblNotes" runat="server"
Text='<%# Eval("notes") %>'></asp:Label>
            </div>
        </div>
    </ItemTemplate>
</asp:Repeater>
```

You also need to set up a style sheet:

```
body
{
```

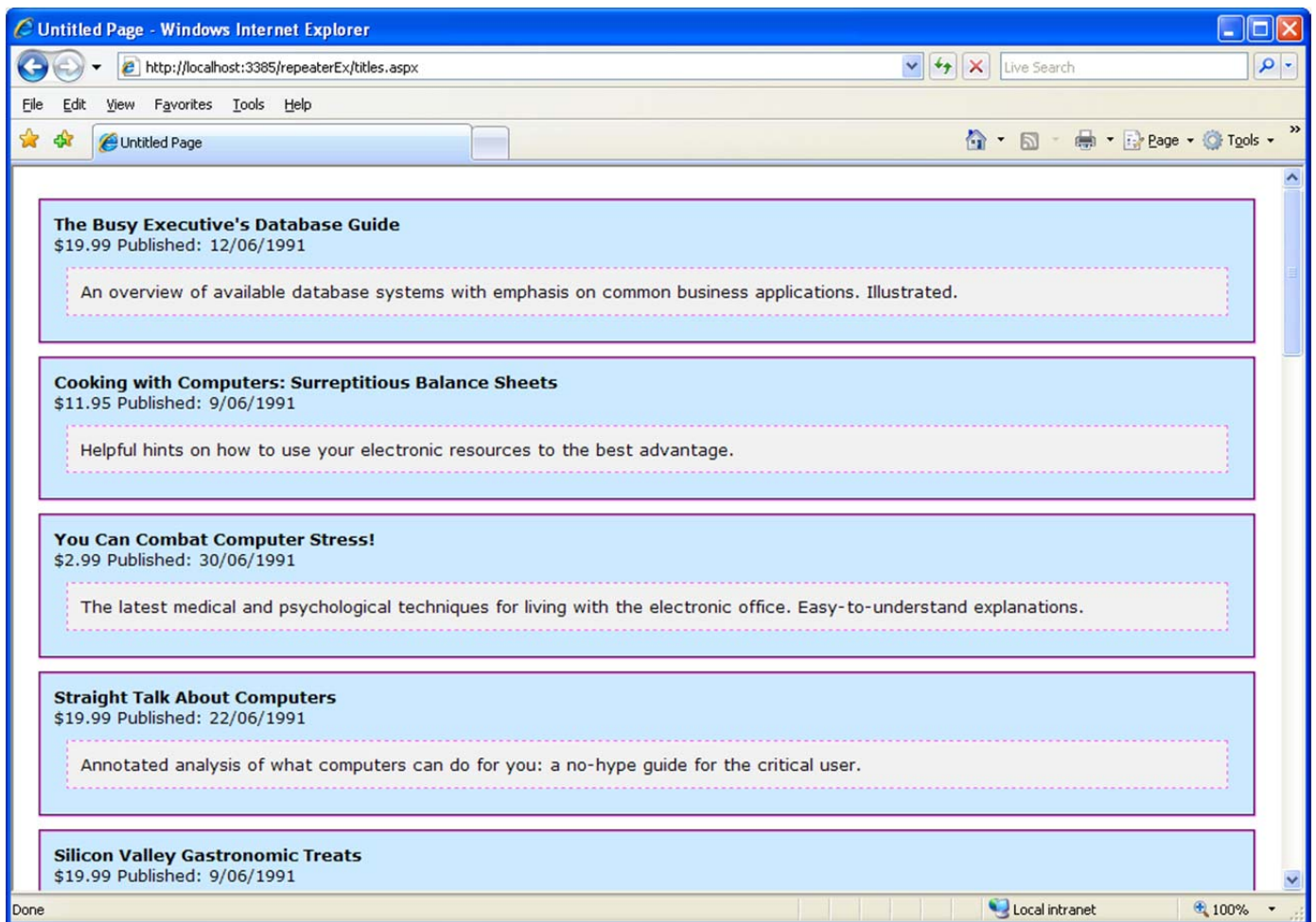
```
font-family: Verdana;
font-size: small;
}

.book
{
    margin: 10px;
    padding: 10px;
    border-style:groove;
    border-color: #FF66FF;
    border-width: 2px;
    background-color: #CCEAFF;
}

.bold
{
    font-weight: bold;
}

.notes
{
    border: dashed 1px #FF66FF;
    padding: 10px;
    margin: 10px;
    background-color: #F1F1F1;
}
```

The screen should look similar to this:



Using templates with the repeater control

The repeater control supports 5 templates:

- ItemTemplate
- AlternatingItemTemplate
- SeparatorTemplate
- HeaderTemplate
- FooterTemplate

Activity 3. Titles for publishers

pubTitles.aspx

In this exercise we will display publishers as hyperlinks along the top of the page. When clicked the title, price and year to date sales for those publishers will be displayed in another repeater control.

The repeater control does not have a selected value property so we need to pass the ID of the publisher as part of the query string.

For this we need 2 data source controls and 2 repeater controls. The first repeater will display the publishers as hyperlinks.

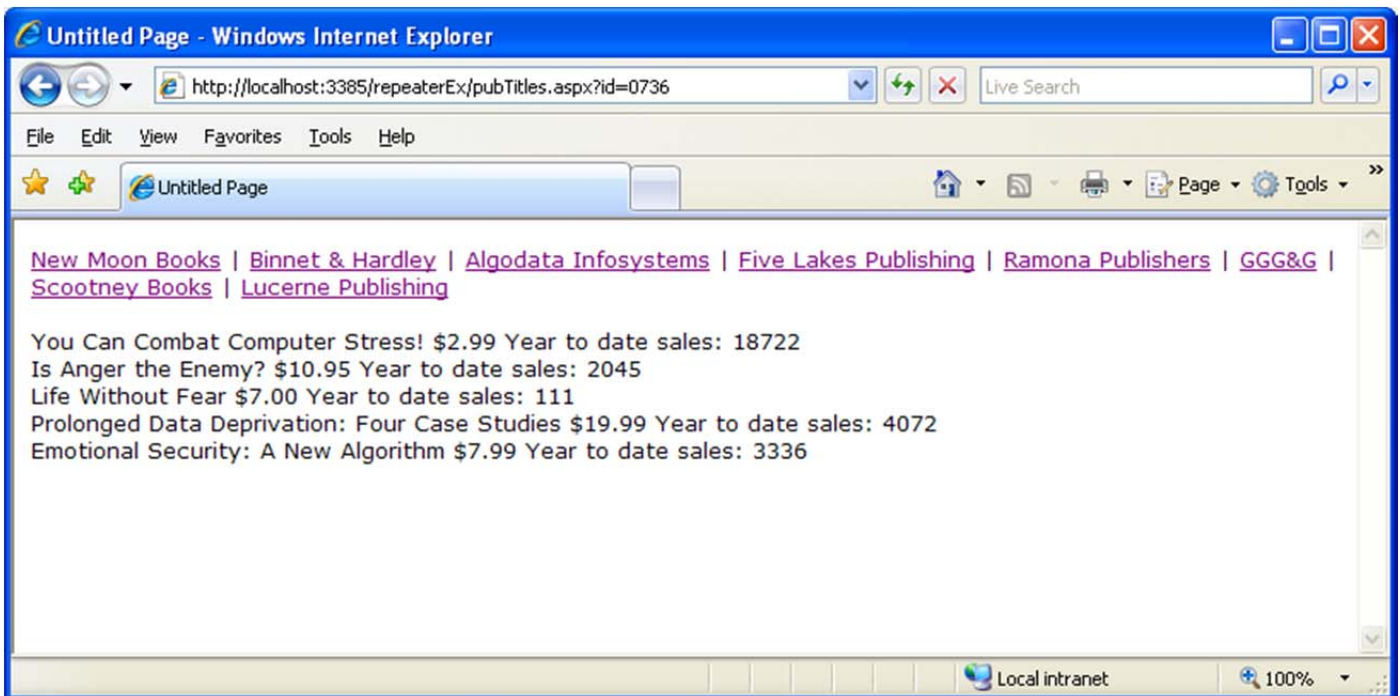
The second repeater will display the title, price and year to date sales for the publisher that the user clicked on.

```
<asp:SqlDataSource ID="sdsPub" runat="server" ConnectionString="<%$
ConnectionStrings:ConnectionStringPubs %>" SelectCommand="SELECT [pub_name], [pub_id]
FROM [publishers]"></asp:SqlDataSource>
<asp:Repeater ID="Repeater1" runat="server" DataSourceID="sdsPub">
<ItemTemplate>
<asp:hyperLink ID="lbPub" runat="server"
Text='<%# Eval("pub_name") %>'
NavigateUrl='<%# Eval("pub_id",
"~/pubTitles.aspx?id={0}") %>'>
</asp:hyperLink>
```

```

</ItemTemplate>
<SeparatorTemplate>
|
</SeparatorTemplate>
</asp:Repeater>
<br />
<asp:SqlDataSource ID="sdsTitles" runat="server" ConnectionString="<%$
ConnectionStrings:ConnectionStringPubs %>" SelectCommand="SELECT [title], [price],
[ytd_sales] FROM [titles] WHERE ([pub_id] = @pub_id)">
<SelectParameters>
<asp:QueryStringParameter Name="pub_id"
QueryStringField="id" Type="String" />
</SelectParameters>
</asp:SqlDataSource>
<br />
<asp:Repeater ID="Repeater2" runat="server" DataSourceID="sdsTitles">
<ItemTemplate>
<asp:Label ID="lblTitle" runat="server"
Text="<%# Eval("title") %>">
</asp:Label>
<asp:Label ID="lblPrice" runat="server"
Text="<%# Eval("price", "{0:c}") %>">
</asp:Label>
Year to date sales:
<asp:Label ID="lblSales" runat="server"
Text="<%# Eval("ytd_sales") %>">
</asp:Label>
</ItemTemplate>
<SeparatorTemplate>
<br />
</SeparatorTemplate>
</asp:Repeater>

```



Handling repeater control events

The repeater control supports the following events:

- **DataBinding** – raised when the repeater control is bound to its data source
- **ItemCommand** – Raised when a control contained in the repeater control raises an event

- ItemCreated – Raised when each repeater item is created
- ItemDataBound – Raised when each repeater item is bound

Activity 4. Wish list

repeaterEvents.aspx

For this exercise we will add a repeater control which will display the title and price of all books.

We will also display a button which when clicked will add the book title to a label on this page this label contains the wish list.

To add the book to the wish list we need to use the itemCommand event as this event is raised whenever a control contained in the repeater control raises an event (i.e. a button is clicked).

Add a data source control to the page and set it up so that it retrieves the title and price from the titles table.

Add a repeater control and display the title and price to labels. Add a button.

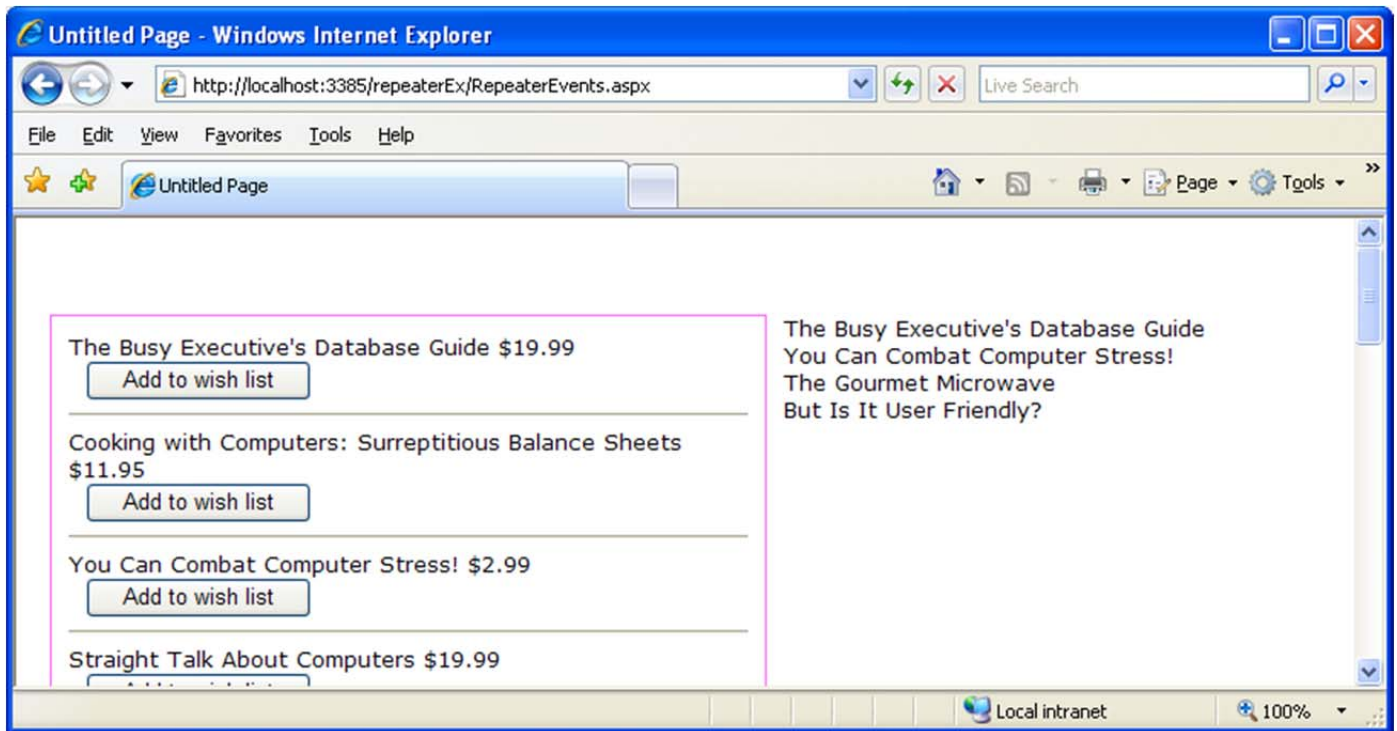
```
<asp:SqlDataSource ID="sdsTitles" runat="server"
ConnectionString="<%$ ConnectionStrings:ConnectionStringPubs %>"
SelectCommand="SELECT [title], [price], [ytd_sales] FROM [titles] ">
</asp:SqlDataSource>
<br />
<div id="bookList">
<asp:Repeater ID="Repeater2" runat="server"
DataSourceID="sdsTitles" onitemcommand="Repeater2_ItemCommand">
<ItemTemplate>
<asp:Label ID="lblTitle" runat="server"
Text="<%# Eval("title") %>"></asp:Label>
<asp:Label ID="lblPrice" runat="server"
Text="<%# Eval("price", "{0:c}") %>"></asp:Label>
<br />
<asp:Button ID="btnAdd" runat="server"
Text="Add to wish list" />
</ItemTemplate>
<SeparatorTemplate>
<hr />
</SeparatorTemplate>
</asp:Repeater>
</div>
<div id="wishList">
<asp:Label ID="lblWish" runat="server"></asp:Label>
</div>
```

Double click on the repeater control in design view to create the ItemCommand event handling method in the code behind. You will also need to add the following code:

```
protected void Repeater2_ItemCommand(object source, RepeaterCommandEventArgs e)
{
    Label lblTempTitle;

    lblTempTitle = (Label)e.Item.FindControl("lblTitle");

    lblWish.Text += lblTempTitle.Text + "<br />";
}
```

References

ASP.NET 2.0 unleashed Chapter 13