# Tutorial 4: Binding Data

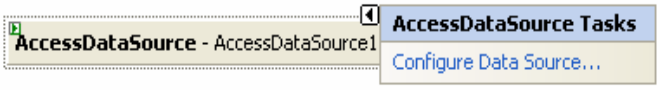
- If you do a great deal of database access in your Web pages, then you will want to make use of a feature known as data binding.
- It's a technique that involves selecting the place(s) where the data comes from (each of which is a data source), and then specifying which fields will be rendered by which controls.
- The rest takes place "automagically," which means your life (that part of it, anyway) becomes pretty easy.

## Creating a data source

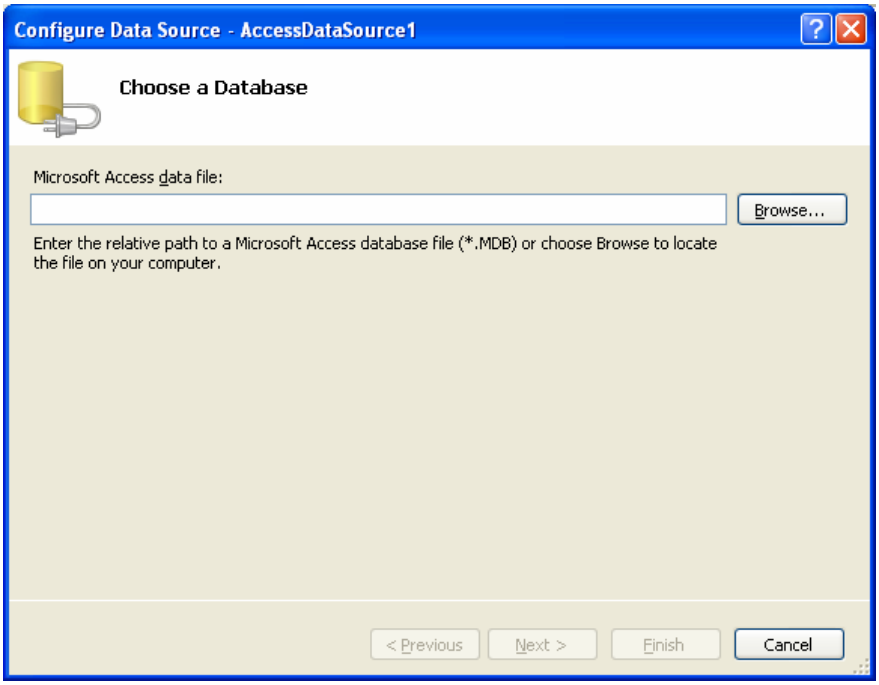
- The different incarnations of Visual Studio.NET provide a wizard for creating data sources that you can then use for supplying data to the controls on your Web page.
- The wizard makes displaying data incredibly easy — but first you have to make sure your database, tables, and users are all set up.
- Otherwise the wizard will fail and you'll have to start over.

To create a data source using the wizard, follow these steps:

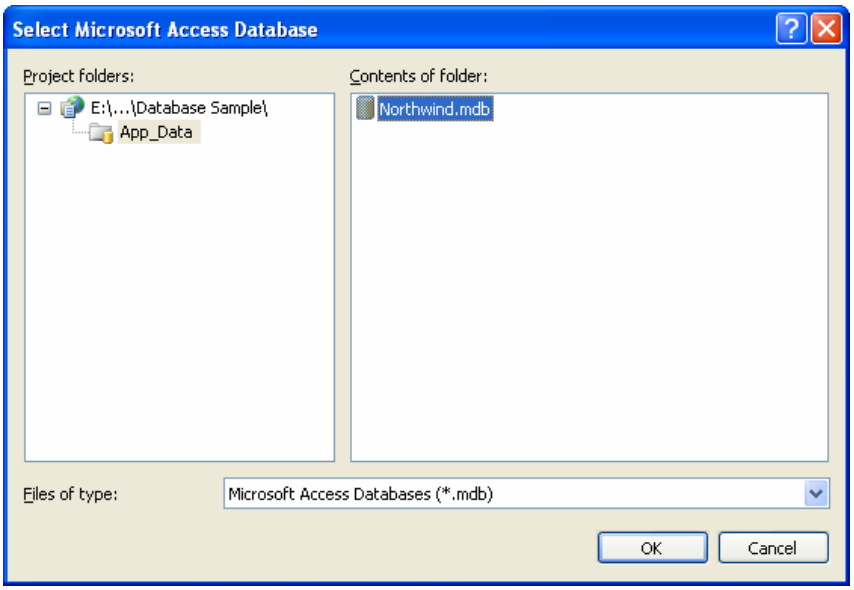
1. Drag the `AccessDataSource` from the toolbox onto a Web form and click the triangle in the upper-right corner.



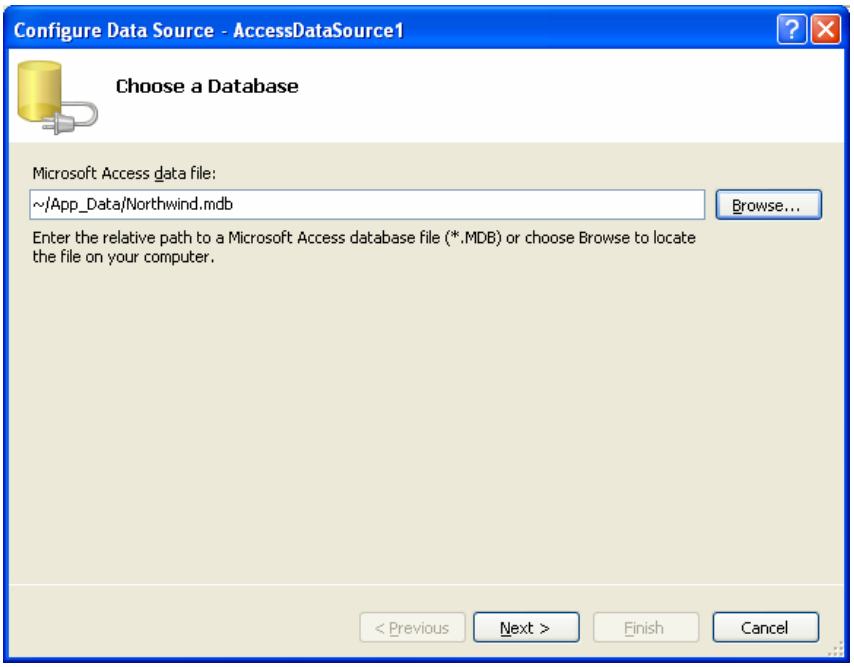

2. In the Smart Tag, click Configure Data Source... to open the Configure Data Source Wizard



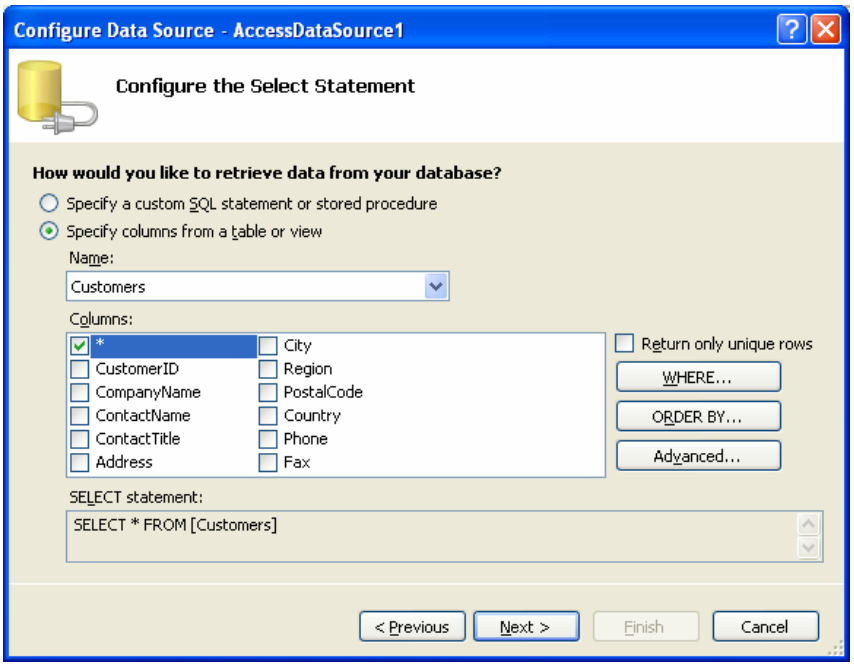
3. In the Configure Data Source Wizard, click the browse...button. This opens the Select Microsoft Access Database dialog box.



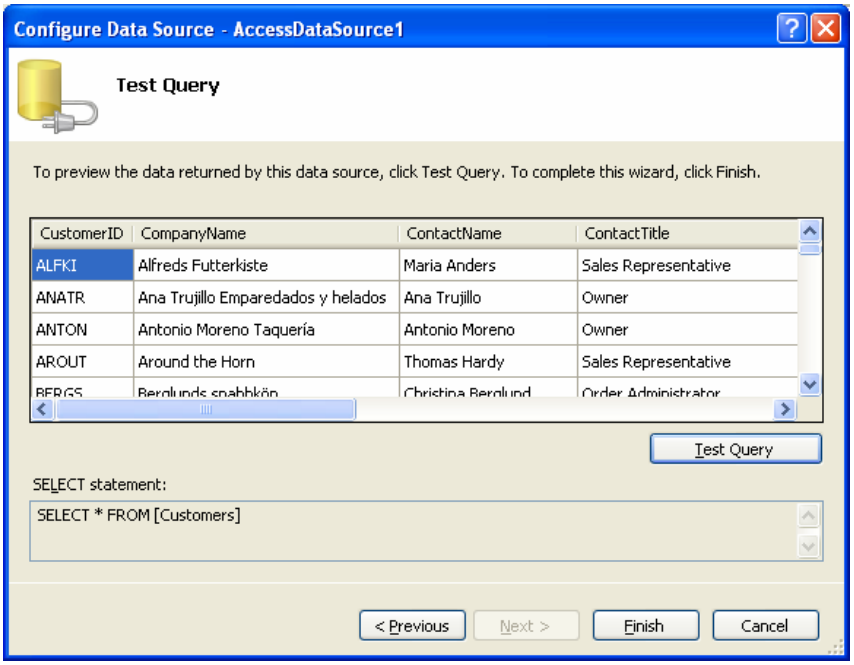

4. In the Select Microsoft Access Database dialog box, select the Northwind database and click OK button. This opens the Configure Data Source dialog box.




5. Click Next in the Configure Data Source Wizard. The page that opens allows you to choose a table.




6. Select the Customers table and click the * in the Columns. Finally, click Next to test out the query. You will see a table that shows the rows from the table

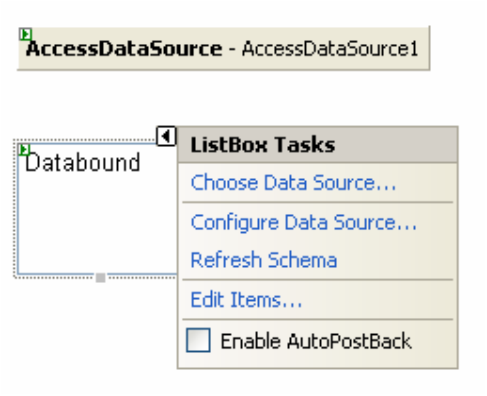


## Using a data source

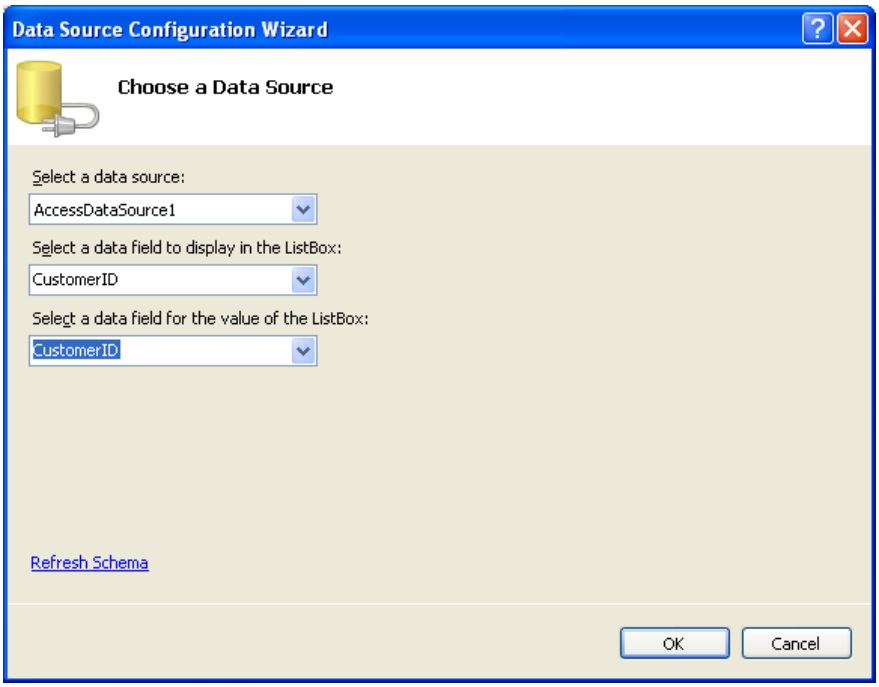
- Once you have a data source created and configured, you can put it to use on your form.
- But first think about one small issue: Your data source is linked to a table, a view, or a stored procedure that returns data. And the data that's returned, though it possesses columns, probably has multiple rows as well.

To show all this data, typically you have to use a control that can show multiple items — for example, a `ListBox`.

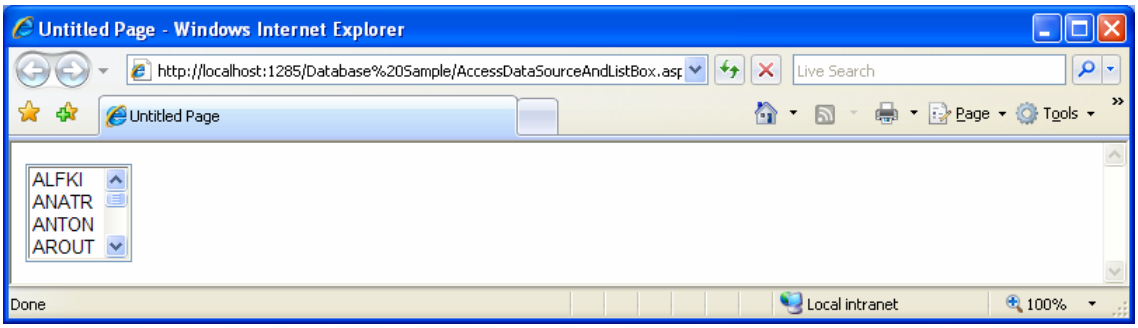
1. Drag the `ListBox` from the toolbox onto a Web form and click the triangle in the upper-right corner.




2. In the Smart Tag, click Choose Data Source... to open the Configure Data Source Configuration Wizard. Select the items as show in the following figure.



3. Here, the `AccessDataSource` control is configured to select all the rows in the Customers table, and the `ListBox` control is configured to display the `CustomerID` column from the result set. Figure below shows what you'll see when you open this page in the browser.

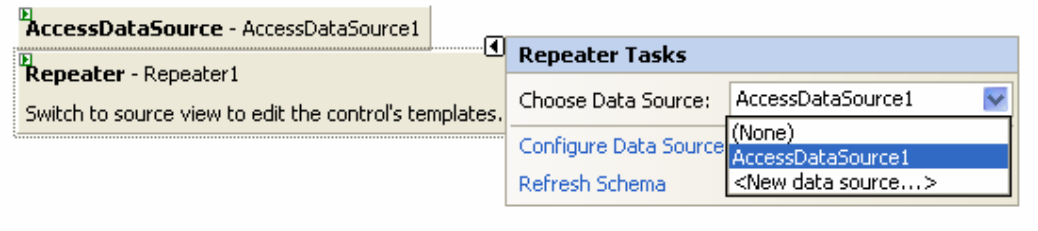



4. Here's some sample .aspx code that displays the data in a `ListBox` control

```
<%@ Page Language="C#" AutoEventWireup="true"
CodeFile="AccessDataSourceAndListBox.aspx.cs"
Inherits="AccessDataSourceAndListBox" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml" >
<head runat="server">
    <title>Untitled Page</title>
</head>
<body>
    <form id="form1" runat="server">
    <div>
        <asp:AccessDataSource ID="AccessDataSource1" runat="server"
            DataFile="~/App_Data/Northwind.mdb"
            SelectCommand="SELECT * FROM [Customers]">
        </asp:AccessDataSource>
        <asp:ListBox ID="ListBox1" runat="server"
            DataSourceID="AccessDataSource1" DataTextField="CustomerID"
            DataValueField="CustomerID"></asp:ListBox> </div>
    </form>
</body>
</html>
```

## Repeating Through a Result Set

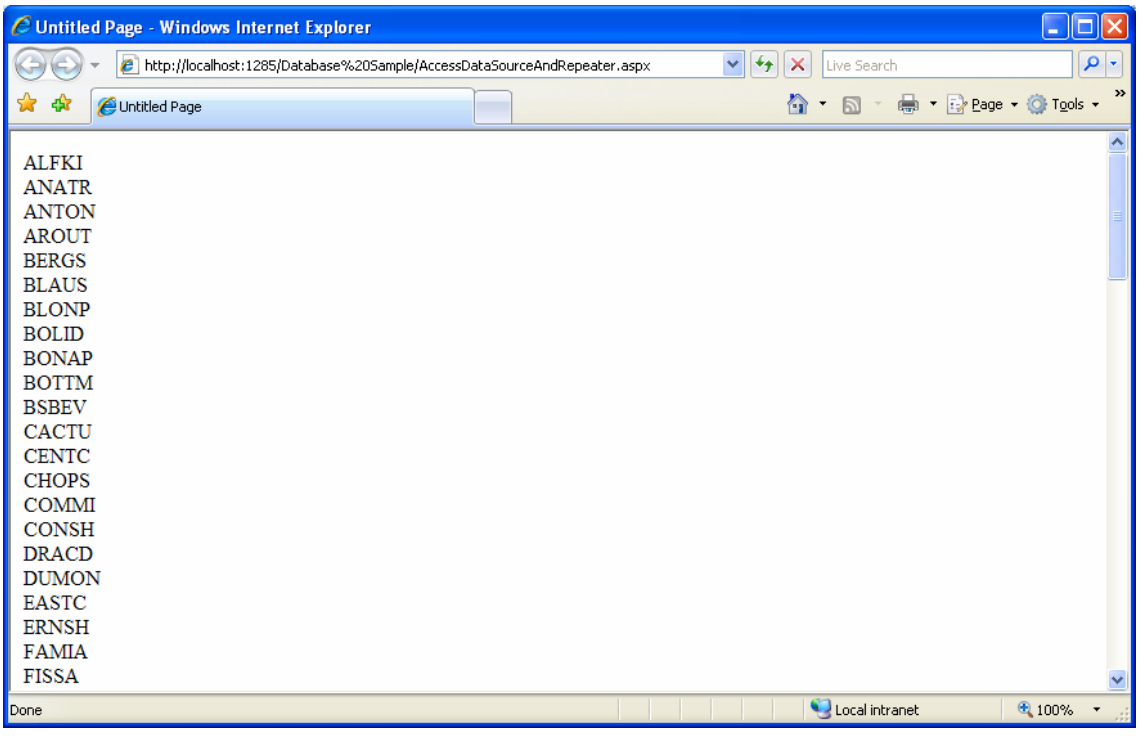
- If you want to display several rows of data and do a lot of custom formatting in your displaying of the data, then the Repeater control is for you.
- The Repeater control is nothing fancy; you simply supply the HTML and ASP.NET controls for each row, and the Repeater control displays your HTML and controls for each row in the data set.
- To do this, just drag the Repeater control from the toolbox onto a Web form and click the triangle in the upper-right corner to select the `AccessDataSource1`.




- Add the following line in to the .aspx file.

```
<asp:Repeater ID="Repeater1" runat="server"
      DataSourceID="AccessDataSource1">
      <ItemTemplate>
            <%# Eval("CustomerID")%>
            <br />
      </ItemTemplate>
</asp:Repeater>
```

- Figure below shows what you'll see when you open this page in the browser.



- Here's some sample .aspx code that displays the data in a `ListBox` control

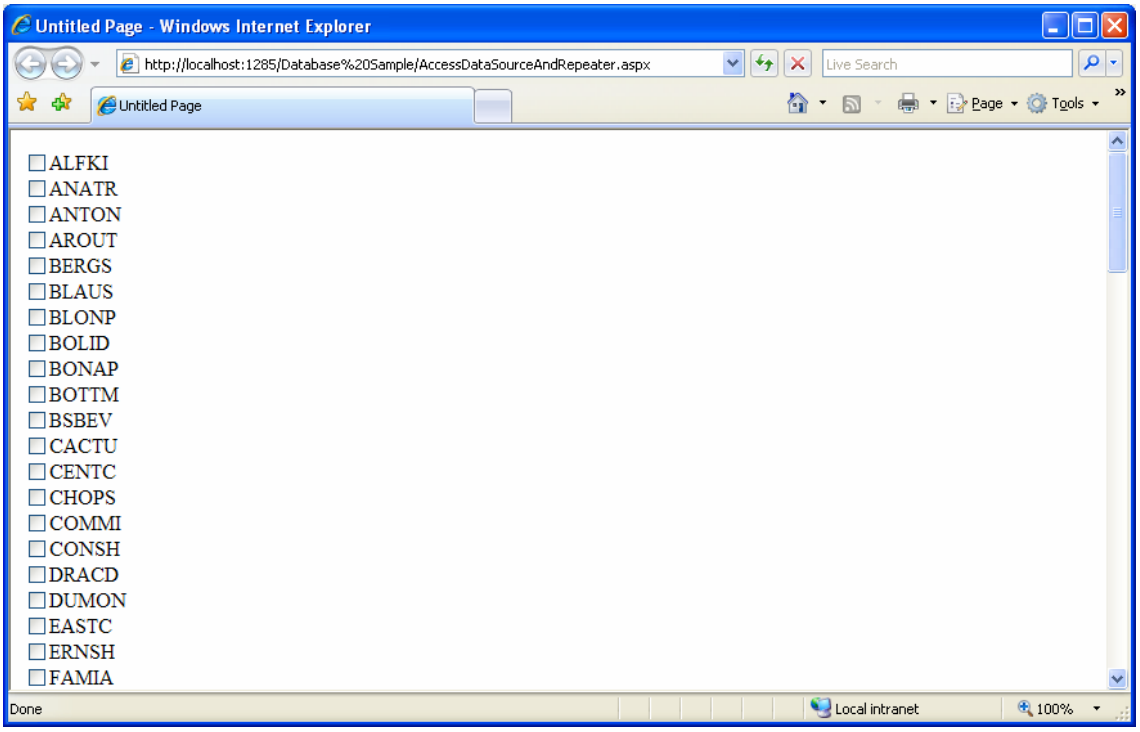
```
<%@ Page Language="C#" AutoEventWireup="true"
CodeFile="AccessDataSourceAndRepeater.aspx.cs"
Inherits="AccessDataSourceAndRepeater" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml" >
<head runat="server">
    <title>Untitled Page</title>
</head>
<body>
    <form id="form1" runat="server">
    <div>
        <asp:AccessDataSource ID="AccessDataSource1" runat="server"
            DataFile="~/App_Data/Northwind.mdb"
            SelectCommand=" SELECT * FROM [Customers]">
        </asp:AccessDataSource>

    </div>
        <asp:Repeater ID="Repeater1" runat="server"
            DataSourceID="AccessDataSource1">
            <ItemTemplate>
                <%# Eval("CustomerID")%>
                <br />
            </ItemTemplate>
        </asp:Repeater>
    </form>
</body>
</html>
```

## Setting server control properties in a Repeater control

- If you put a server control inside a Repeater control, setting its properties to values from the data is easy.
- You just use a binding expression to set the property.
- Take a look at the following code to see how this works:

```
<asp:Repeater ID="Repeater1" runat="server"
      DataSourceID="AccessDataSource1">
      <ItemTemplate>
            <asp:CheckBox runat="server" Text=
                  <%# Eval("CustomerID")%>/>
            <br />
      </ItemTemplate>
</asp:Repeater>
```

- Figure below shows what you'll see when you open this page in the browser.

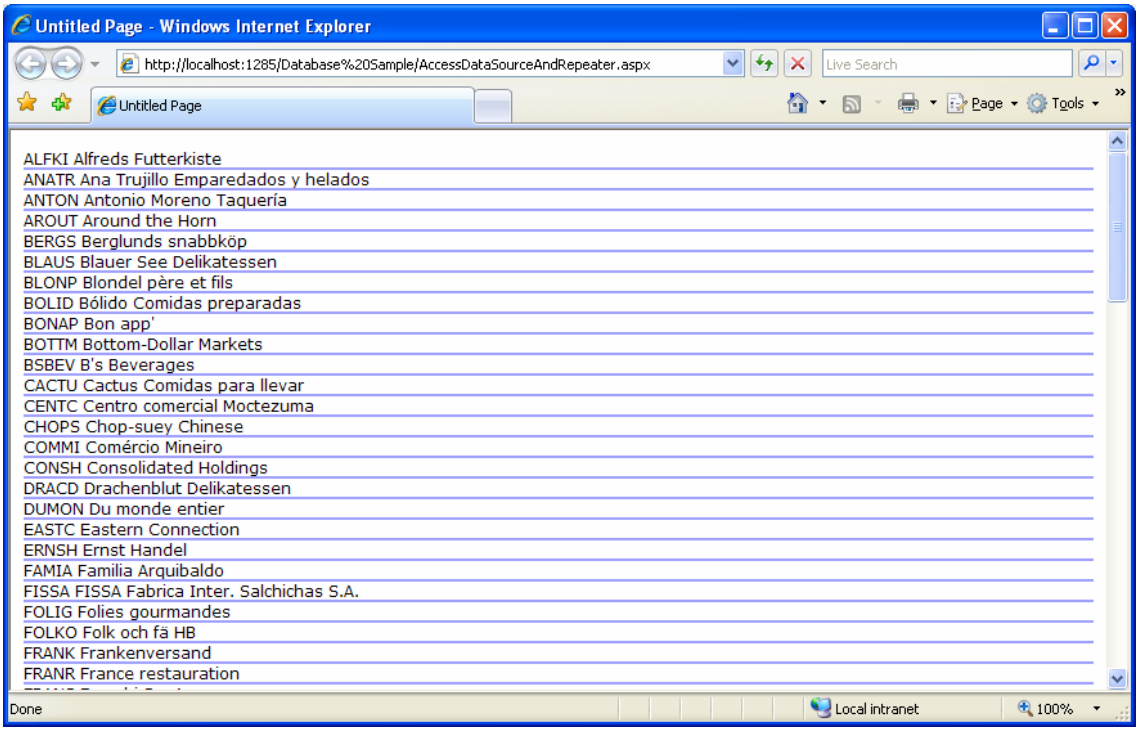



## Adding some style to the Repeater control

- With the basics of the Repeater control under your belt, you can easily make each row in the Repeater control show as many columns as you want in your data set, and you can format the lines as you want.
- Take a look at this example:

```
<asp:Repeater ID="Repeater1" runat="server"
      DataSourceID="AccessDataSource1">
      <ItemTemplate>
            <div style="font-family:Verdana;font-size:12px;
                  border-bottom:solid 2px #A0A0FF">
                  <%# Eval("CustomerID")%>
                  <%# Eval("CompanyName")%>
            </div>
      </ItemTemplate>
</asp:Repeater>
```

- Figure below shows what you'll see when you open this page in the browser.




## Altering the look of the Repeater control's rows

- When you're displaying lots of data, sometimes reading the data can be difficult on the eyes.
- You might want to provide an alternating style, such as alternating background or alternating font color from row to row.
- Doing so helps make the data more readable (or just nicer to look at on-screen).
- The Repeater control provides two different templates, one for the first line and all odd-numbered lines, and one for all the other lines.
- The one for the first and all odd-numbered lines is the standard template that you normally use – `ItemTemplate`.

- For the other lines, you use a template called `AlternatingItemTemplate`. Here's an example:

```
<asp:Repeater ID="Repeater1" runat="server"
      DataSourceID="AccessDataSource1">
      <ItemTemplate>
            <div style="font-family:Verdana;font-size:12px;
                  background-color:#A0A0FF">
                  <%# Eval("CustomerID")%>
                  <%# Eval("CompanyName")%>
            </div>
      </ItemTemplate>
      <AlternatingItemTemplate>
            <div style="font-family:Verdana;font-size:12px;
                  background-color:#FFFFFF">
                  <%# Eval("CustomerID")%>
                  <%# Eval("CompanyName")%>
            </div>
      </AlternatingItemTemplate>
</asp:Repeater>
```

- There are two templates inside the `asp:Repeater` tag.
- The first is `ItemTemplate`, and the second is `AlternatingTemplate`.
- The only difference between them is the style property of the div tag — the first one has a
- bluish background color, and the second has a white background color.
- The on-screen result is alternating dark and light lines, as shown in the following figure.