Databinding to an Accordion

Christian Wenz

Overview

The **ReorderList** control in the AJAX Control Toolkit provides a list that can be reordered by the user via drag and drop. The current order of the list shall be persisted on the server.

Steps

The **ReorderList** control supports binding data from a database to the list. Best of all, it also supports writing changes to the order of the list element back to the data store. This sample uses Microsoft SQL Server 2005 Express Edition as the data store. The database is an optional (and free) part of a Visual Studio installation, including express edition. It is also available as a separate download under http://go.microsoft.com/fwlink/?LinkId=64064. For this sample, we assume that the instance of the SQL Server 2005 Express Edition is called **SQLEXPRESS** and resides on the same machine as the web server; this is also the default setup. If your setup differs, you have to adapt the connection information for the database.

The easiest way to set up the database is to use the Microsoft SQL Server Management Studio Express (http://www.microsoft.com/downloads/details.aspx?FamilyID=c243a5ae-4bd1-4e3d-94b8-5a0f62bf7796&DisplayLang=en). Connect to the server, double-click on **Databases** and create a new database (right-click and choose **New Database**) called **Tutorials**.

In this database, create a new table called **AJAX** with the following four columns:

- **id** (primary key, integer, identity, not NULL)
- **char** (char(1), NULL)
- **description** (varchar(50), NULL)
- **position** (int, NULL)
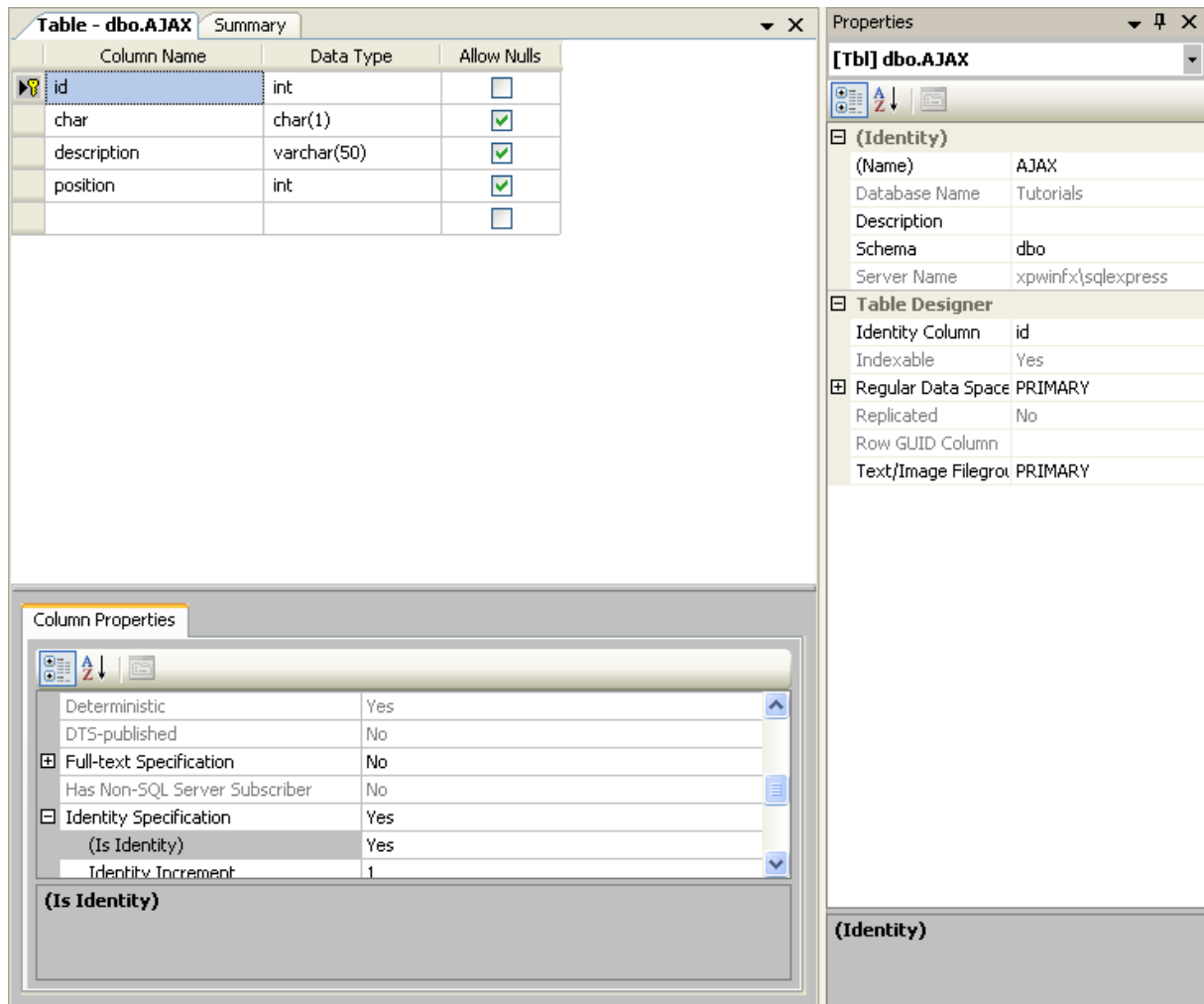
The layout of the AJAX table

Next, fill the table with a couple of values. Note that the **position** column holds the sort order of the elements.

The initial data in the AJAX table

The next step requires to generate an **SqlDataSource** control to communicate with the new database and its table. The data source must support the **SELECT** and **UPDATE** SQL commands. When the order of the list elements is later changed, the **ReorderList** control automatically submits two values to the data source's **Update** command: the new position and the ID of the element. Therefore, the data source needs an **<UpdateParameters>** section for these two values:

```
<asp:SqlDataSource ID="sds" runat="server" ConnectionString="Data
  Source=(local)\SQLEXPRESS;Initial Catalog=Tutorials;Integrated
  Security=True"
  ProviderName="System.Data.SqlClient"
  OldValuesParameterFormatString="original_{0}"
  SelectCommand="SELECT [id], [char], [description], [position]
  FROM [AJAX] ORDER BY [position]"
  UpdateCommand="UPDATE [AJAX] SET position=@position WHERE
  [id]=@original_id">
  <UpdateParameters>
    <asp:Parameter Name="position" Type="Int32" />
    <asp:Parameter Name="original_id" Type="Int32" />
  </UpdateParameters>
</asp:SqlDataSource>
```

The **ReorderList** control needs to set the following attributes:

- **AllowReorder**: Whether the list items may be rearranged
- **DataSourceID**: The ID of the data source
- **DataKeyField**: The name of the primary key column in the data source
- **SortOrderField**: The data source column that provides the sort order for the list items

In the **<DragHandleTemplate>** and **<ItemTemplate>** sections, the layout of the list can be fine-tuned. Also, databinding is possible using the **Eval()** method, as seen here:

```
<ajaxToolkit:ReorderList ID="rl1" runat="server"
  SortOrderField="position" AllowReorder="true"
  DataSourceID="sds" DataKeyField="id">
  <DragHandleTemplate>
    <div class="DragHandleClass">
    </div>
  </DragHandleTemplate>
  <ItemTemplate>
      <asp:Label ID="ItemLabel" runat="server"
  Text='<%#Eval("description") %>' />
  </ItemTemplate>
</ajaxToolkit:ReorderList>
```

The following CSS style information (referenced in the **<DragHandleTemplate>** section of the **ReorderList** control) makes sure that the mouse pointer changes appropriately when it hovers over the drag handle:

```
<style type="text/css">
  .DragHandleClass {
    width: 12px;
    height: 12px;
    background-color: red;
    cursor:move;
  }
</style>
```

Finally, a **ScriptManager** control initializes ASP.NET AJAX for the page:

```
<asp:ScriptManager ID="asm" runat="server" />
```

Run this example in the browser and rearrange the list items a bit. Then, reload the page and/or have a look at the database. The altered positions have been maintained and are also reflected by the values in the **position** column in the database—and that all without any code, just by using markup.

id	char	description	position
1	A	Asynchronous	3
2	J	JavaScript	4
3	A	And	2
4	X	XML	1
NULL	NULL	NULL	NULL

Table - dbo.AJAX / Summary

The data in the database changes according to the new list item order