Custom Templates for AJAX Grids

Data Aquarium Framework and Aquarium Express provide excellent support for custom form templates.

New feature in Data Aquarium Framework allows to create *inline form templates* for rows of grid views in edit mode.

See live demo of this feature now.

Generate a *Data Aquarium* project from *Northwind* database. Create a new form with name *GridTemplate.aspx* based on *MasterPage.master*. Enter the following markup.

```
<%@ Page Language="VB" MasterPageFile="~/MasterPage.master"
AutoEventWireup="false"
    CodeFile="GridTemplate.aspx.vb" Inherits="GridTemplate" Title="Grid
Template" %>


<asp:Content ID="Content1" ContentPlaceHolderID="head" runat="Server">
</asp:Content>
<asp:Content ID="Content2" ContentPlaceHolderID="Header1Placeholder"
runat="Server">
    Grid Template
</asp:Content>
<asp:Content ID="Content3" ContentPlaceHolderID="Header2Placeholder"
runat="Server">
    Northwind
</asp:Content>
<asp:Content ID="Content4" ContentPlaceHolderID="BodyPlaceholder"
runat="Server">
    <div id="Products" runat="server" />
    <aquarium:DataViewExtender ID="ProductsExtender" runat="server"
Controller="Products"
        TargetControlID="Products" />
```

```
</asp:Content>
```

View this page in a web browser. It will look like the one in the next illustration.
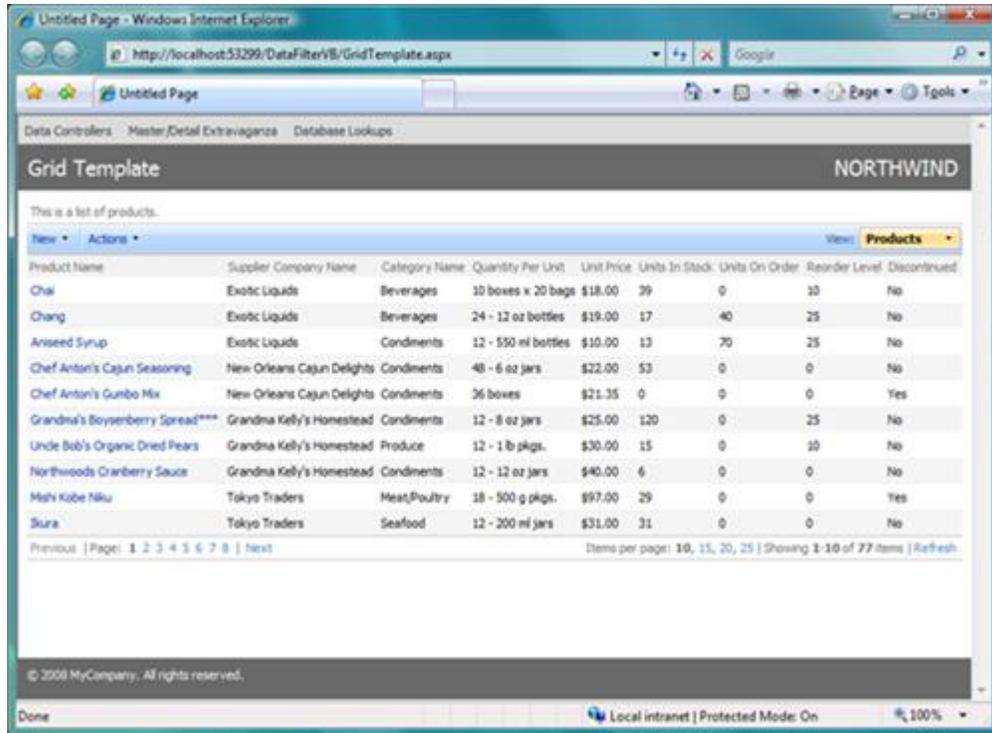
Change *~/Controllers/Products.aspx* data controller descriptor by moving action with *Edit* command to be the first action in the scope of grid.

```
<actionGroup scope="Grid">
  <action commandName="Edit" />
  <action commandName="Select" commandArgument="editForm1" />
  <action commandName="Delete" confirmation="Delete?" />
  <action whenLastCommandName="Edit" commandName="Update" headerText="Save"
/>
  <action whenLastCommandName="Edit" commandName="Cancel" />
</actionGroup>
```

Refresh the page and click on any link in the first column. This will force the grid view to display all fields in selected row to present the edit mode user interface.
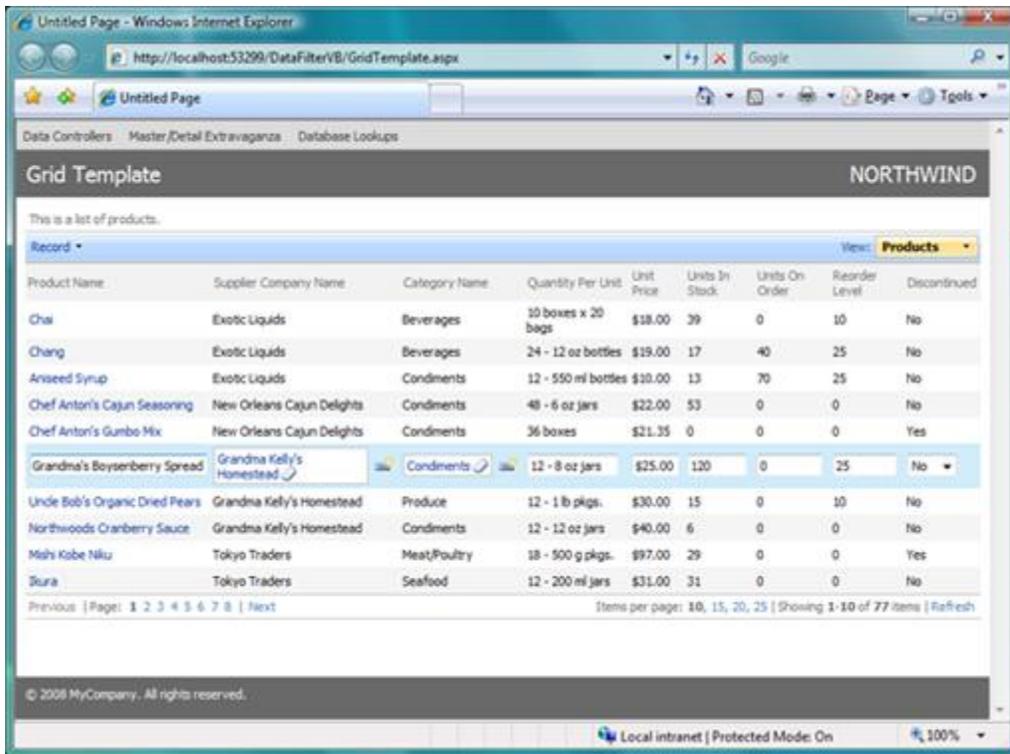
This works well for most applications. Sometimes you may need to display just a few columns when user browses through records but would rather have more fields available for editing without leaving the grid view.

Enter the following *HTML* fragment right after the *DataViewExtender* in *GridTemplate.aspx* page.

```
<div id="Products_grid1" style="display: none">
    <table style="border: solid 1px silver; margin: 4px">
        <tr>
            <td valign="top">
                <div style="margin-bottom: 4px">
                    {ProductName}</div>
                <div style="margin-bottom: 4px">
                    {SupplierID}</div>
                <div>
                    {CategoryID}</div>
            </td>
```

```
<td valign="top">
    <div style="margin-bottom: 4px">
        {QuantityPerUnit}</div>
    <div style="margin-bottom: 4px">
        {UnitPrice}</div>
    <div>
        {Discontinued}</div>
</td>
<td valign="top">
    <div style="margin-bottom: 4px">
        {UnitsInStock}</div>
    <div style="margin-bottom: 4px">
        {UnitsOnOrder}</div>
    <div>
        {ReorderLevel}</div>
</td>
    </tr>
    </table>
</div>
```

Refresh the page in a web browser and click on any product link in the first column of the grid view.
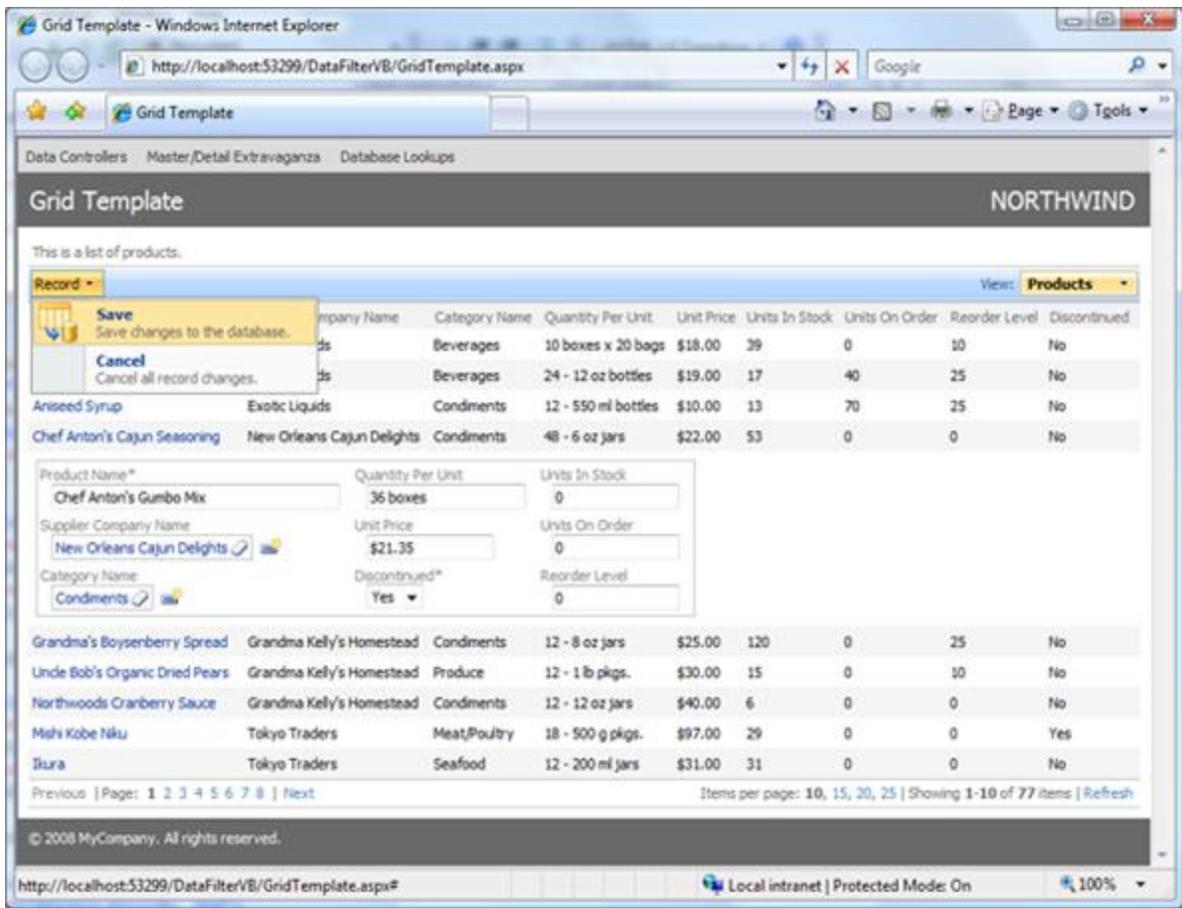
Your template will be displayed in place of the selected row. The template definition technique is equivalent to the one that you would use with custom form templates. The popup menu on the action bar allows user to save or cancel changes. All controls behave in the same way as they do in a regular form view rendered by framework.

From user's perspective such grid is a lot like an accordion view, which expands the row when you edit the record. You can further enhance the presentation of inline form by introducing custom *Save* and *Cancel* buttons. Add a second table row to the table in the template *Products_grid1*.

```
<tr>
    <td colspan="3" align="right" style="border-top: solid 1px silver;
padding: 4px;">
        <button
onclick="Web.DataView.find('ProductsExtender').executeAction('ActionBar', 0,
null, 0)">
            Save</button>
        <button
onclick="Web.DataView.find('ProductsExtender').executeAction('ActionBar', 1,
null, 0)">
            Close</button>
    </td>
</tr>
```

The novelty here is *onclick* event handler that executes the same actions that are available in the *Record* menu on action bar. This code is written in java script and is executed in a browser when user clicks on a button.

The first argument of *executeAction* method defines the action scope. The second one is the action index. Third argument represents the row in the grid. You don't know, which row is selected, but *Web.DataView* component does. That is why we are passing *null* here. The last argument defines the index of action group in the scope, which is handy if you have more than one action available on action bar in edit mode.

This is how you template looks if you are trying it out.

Notice that this sample shows an invalid *Units On Order* value entered, which is detected when you click on *Save* button.

If you want to display more fields then there are columns in the view then make sure to define all data fields in grid view and simply add *hidden* attribute set to *true* for each data field, which is not supposed to be displayed as a grid view column but should be available in an inline form.