



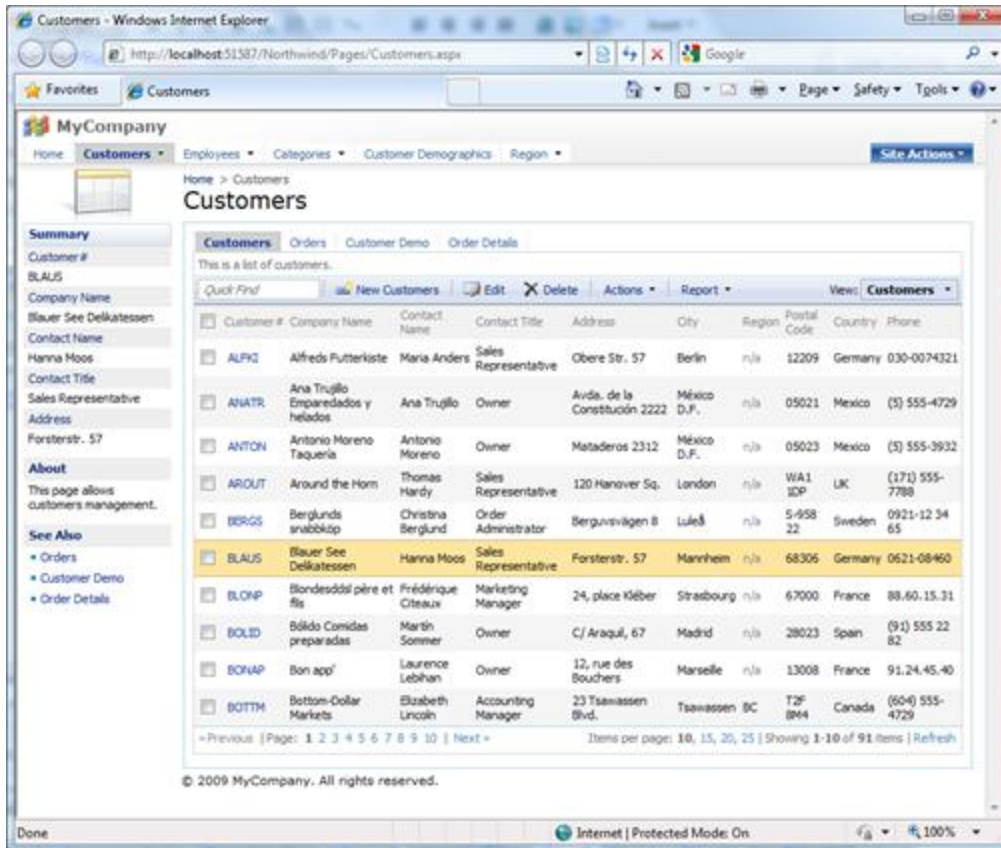
Row Selection Mode

Component *DataViewExtender* allows control over selection mode of rows displayed in data controller grid views via *SelectionMode* property. The default property value is *Single*.

Here is the snippet of *DataViewExtender* definition from *Customers.aspx* page generated with [Web Site Factory](#) premium project from *Northwind* database. The *SelectionMode* property is set to *Multiple*.

```
<div factory:activator="Tab|Customers">
  <div id="view1" runat="server">
    </div>
    <aquarium:DataViewExtender ID="view1Extender" runat="server"
      TargetControlID="view1" Controller="Customers" View="grid1"
      ShowInSummary="true" SelectionMode="Multiple" />
  </div>
```

Here is how the page is displayed in a web browser.

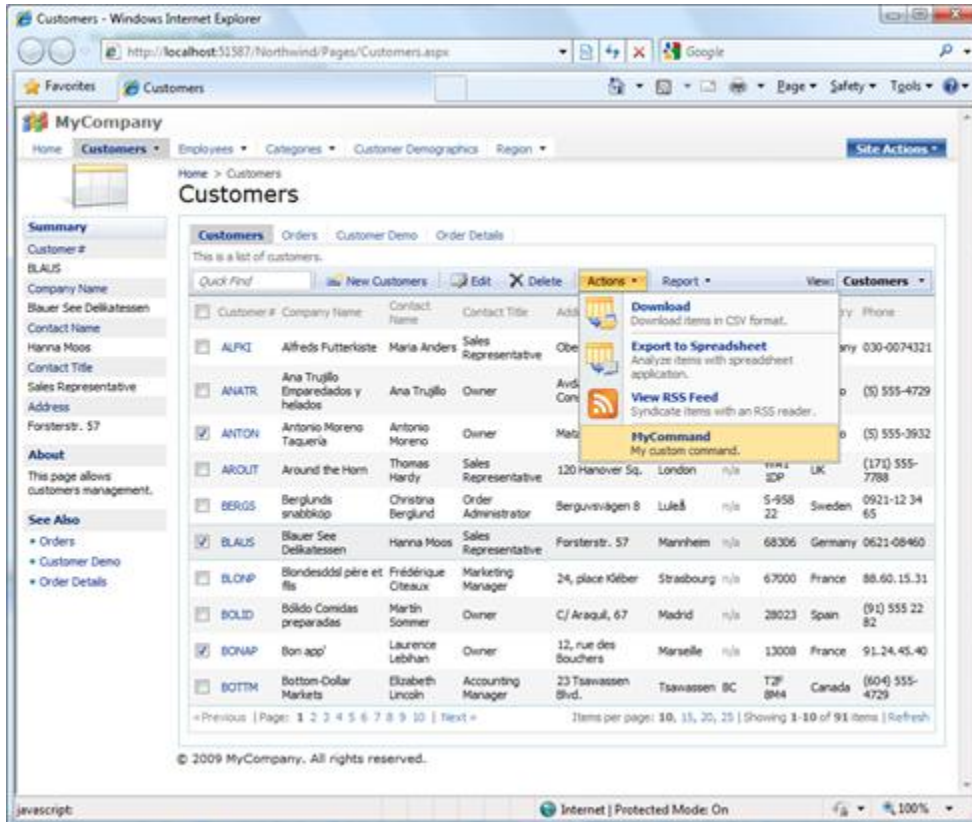


We will also change the data controller `~/Controllers/Customers.xml` to expose a new action `a6` with command argument `MyCommand` on the action bar.

```
<actionGroup id="ag5" scope="ActionBar" headerText="Actions">
  <action id="a1" commandName="ExportCsv" headerText="Download"
    description="Download items in CSV format." />
  <action id="a2" />
  <action id="a3" commandName="ExportRowset" headerText="Export to
Spreadsheet"
  description="Analyze items with spreadsheet<br/> application."
/>
  <action id="a4" commandName="ExportRss" headerText="View RSS Feed"
  description="Syndicate items with an RSS reader." />
  <action id="a5"/>
  <action id="a6" commandName="Custom" commandArgument="MyCommand"
  headerText="MyCommand" description="My custom command."/>
</actionGroup>
```

</actionGroup>

You can see the action displayed as a menu option on action bar of the grid view *grid1*.



Let's add a business rule class *Class1* and link it to the `~/Customers/Customers.xml` data controller via `/dataController/@handler` attribute as described at <http://blog.codeontime.com/2009/04/filtering-and-business-rules.html> and <http://blog.codeontime.com/2009/04/externalfilter-and-modal-views.html>.

```
using System;
using System.Collections.Generic;
using System.Web;
using MyCompany.Data;

public class Class1: BusinessRules
{
```

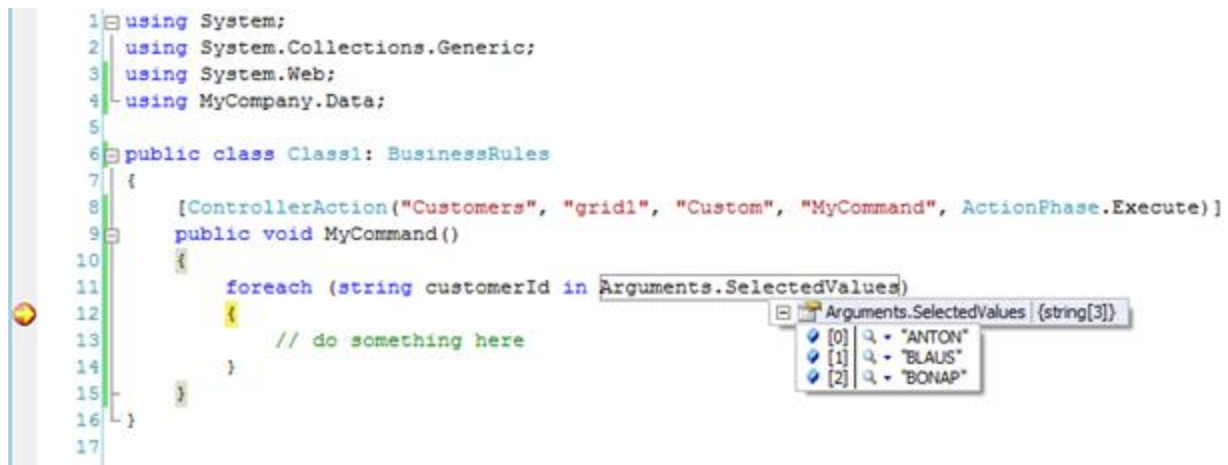
```

        [ControllerAction("Customers", "grid1", "Custom", "MyCommand",
ActionPhase.Execute)]
    public void MyCommand()
    {
        foreach (string customerId in Arguments.SelectedValues)
        {
            // do something here
        }
    }
}

```

Property *SelectedValues* is an array of string values. You might need to convert each value to the appropriate data type when writing custom business rules.

Set the breakpoint inside of *foreach* loop, open the page in a web browser, select a few records, and execute the action. The Visual Studio debugger will show the selected values as pictured below.



Code OnTime LLC

<http://www.codeontime.com>