

2011

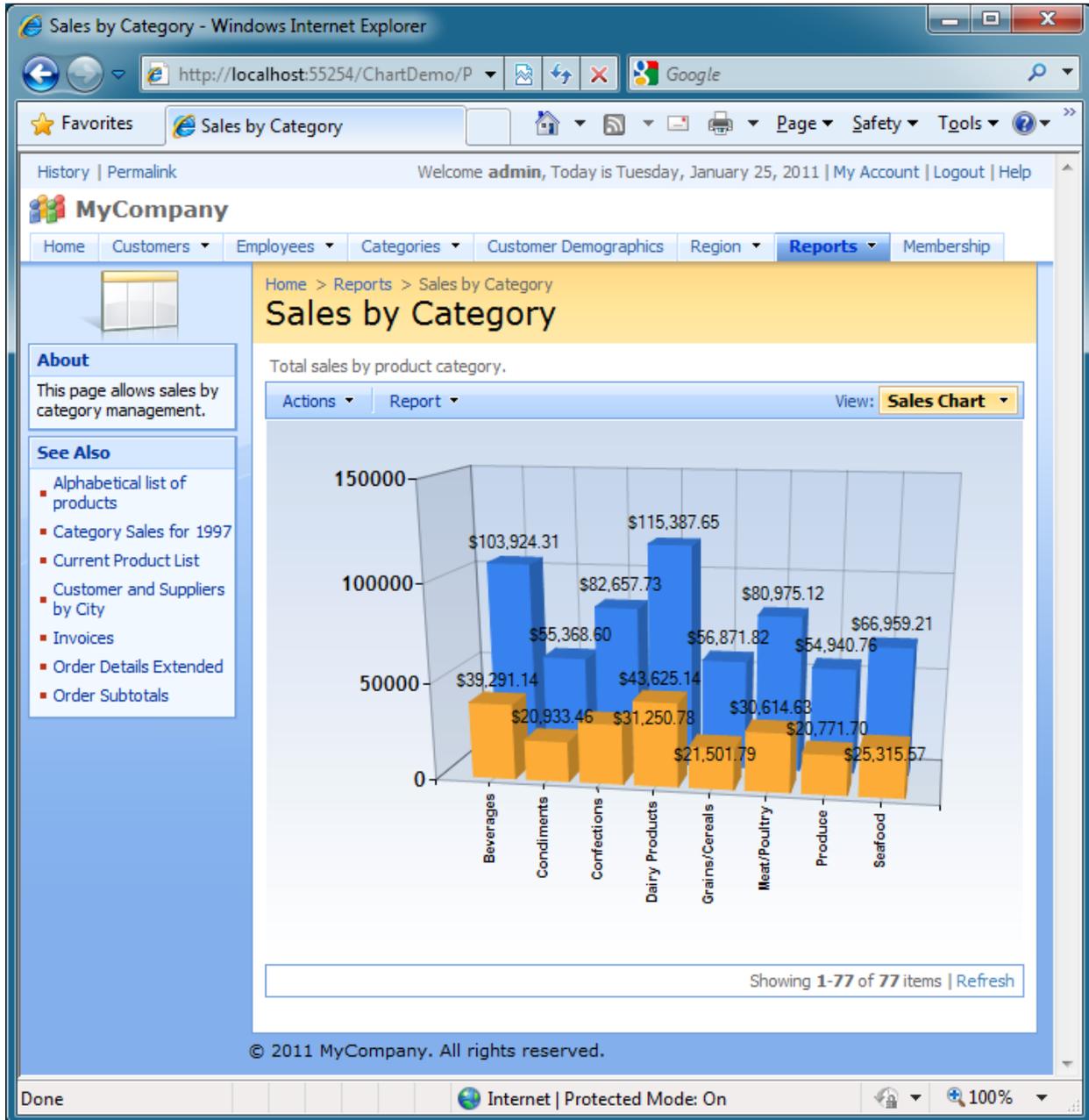


COOKBOOK

Chart View

Chart View

Code On Time applications now offer a new type of data rendering – “Chart” view. Chart view is just another way of presenting a set of data records retrieved from the database. Chart view supports many end-user features including sorting and adaptive filtering.



Creating a Chart View

Generate a new web application from the Northwind database. Browse the generated web site and select *Reports | Sales by Category* menu option. The following data view will be displayed.

Columns *CategoryID*, *Category Name*, *Product Name*, and *Product Sales* are visible in the grid view.

The screenshot shows a web browser window displaying a report titled "Sales by Category". The report is presented as a table with the following data:

Category #	Category Name	Product Name	Product Sales
1	Beverages	Chai	\$4,887.00
1	Beverages	Chang	\$7,038.55
1	Beverages	Chartreuse	\$4,475.70
1	Beverages	Côte de Blay	\$49,198.09
1	Beverages	Guaraná Far	\$1,630.13
1	Beverages	Ipoh Coffee	\$11,069.90
1	Beverages	Lakkalikööri	\$7,379.10
1	Beverages	Laughing Lu	\$5,468.40
1	Beverages	Outback Lag	\$4,485.54
1	Beverages	Rhönbräu Kl	

The data controller is based on the database view *dbo.[Sales by Category]*. This view is a part of the Northwind database and is defined as follows.

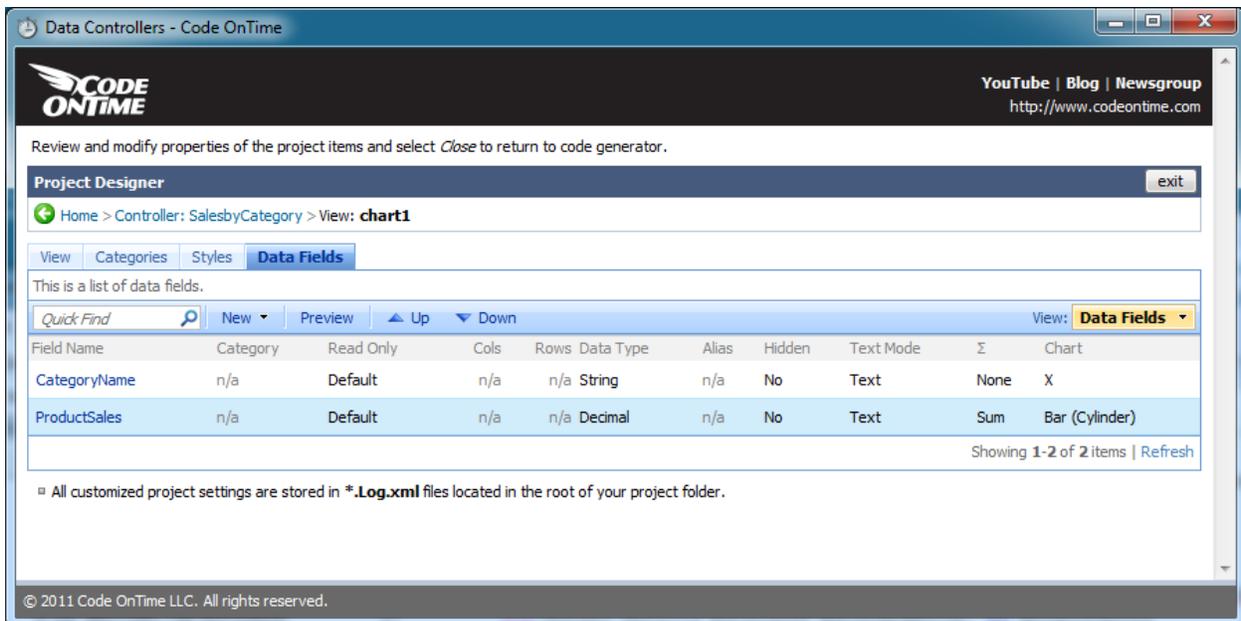
```
create view [dbo].[Sales by Category] AS
SELECT Categories.CategoryID, Categories.CategoryName, Products.ProductName,
       Sum("Order Details Extended".ExtendedPrice) AS ProductSales
FROM   Categories INNER JOIN
       (Products INNER JOIN
        (Orders INNER JOIN "Order Details Extended" ON
         Orders.OrderID = "Order Details Extended".OrderID)
        ON Products.ProductID = "Order Details Extended".ProductID)
       ON Categories.CategoryID = Products.CategoryID
WHERE  Orders.OrderDate BETWEEN '19970101' And '19971231'
GROUP BY Categories.CategoryID, Categories.CategoryName, Products.ProductName
```

Start the code generator, select the project name, and click *Design* button. Select the data controller *SalesbyCategory* and click on *Views* tab.

Add a new view, set its *Id* to *chart1*, select *Chart* as view type, and select *command1* as command. Set label to *Sales Chart*. Enter “*Total sales by product category.*” in the header text.

Save the view and click on its name in the list of available data controller views, select *Data Fields* tab. Add new data field with the field name set to *CategoryName*. Set its *Chart* property under *Miscellaneous* section to *X*. Save the field.

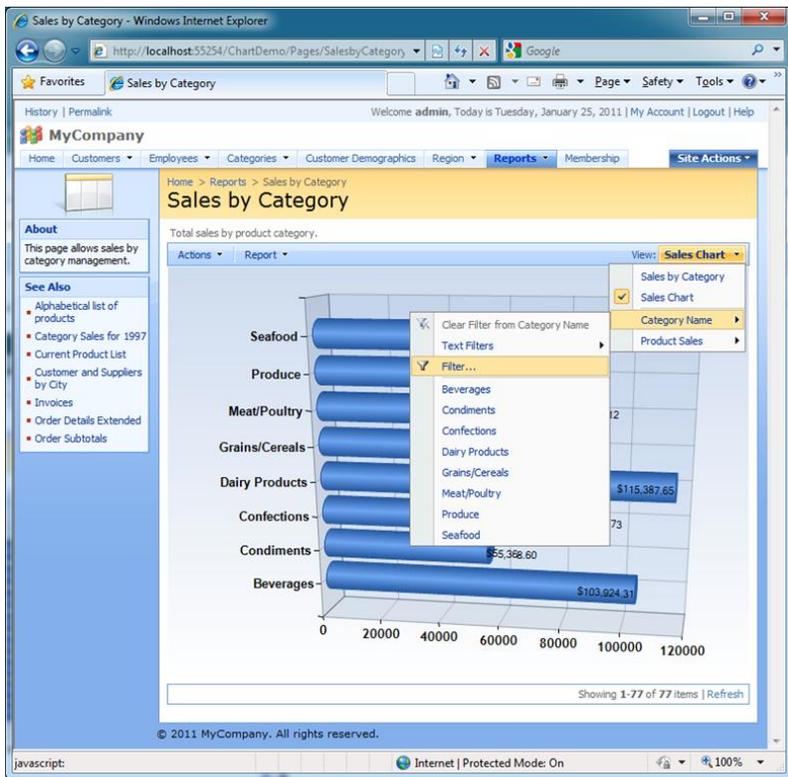
Add another data field with the field name set to *ProductSales*. Enter letter “*c*” without double quotes into *Data Format String*. Set the *Aggregate Function* property of the data field to *Sum*. Set its *Chart* property to *Bar (Cylinder)*. The list of data views in *Designer* will look as follows.



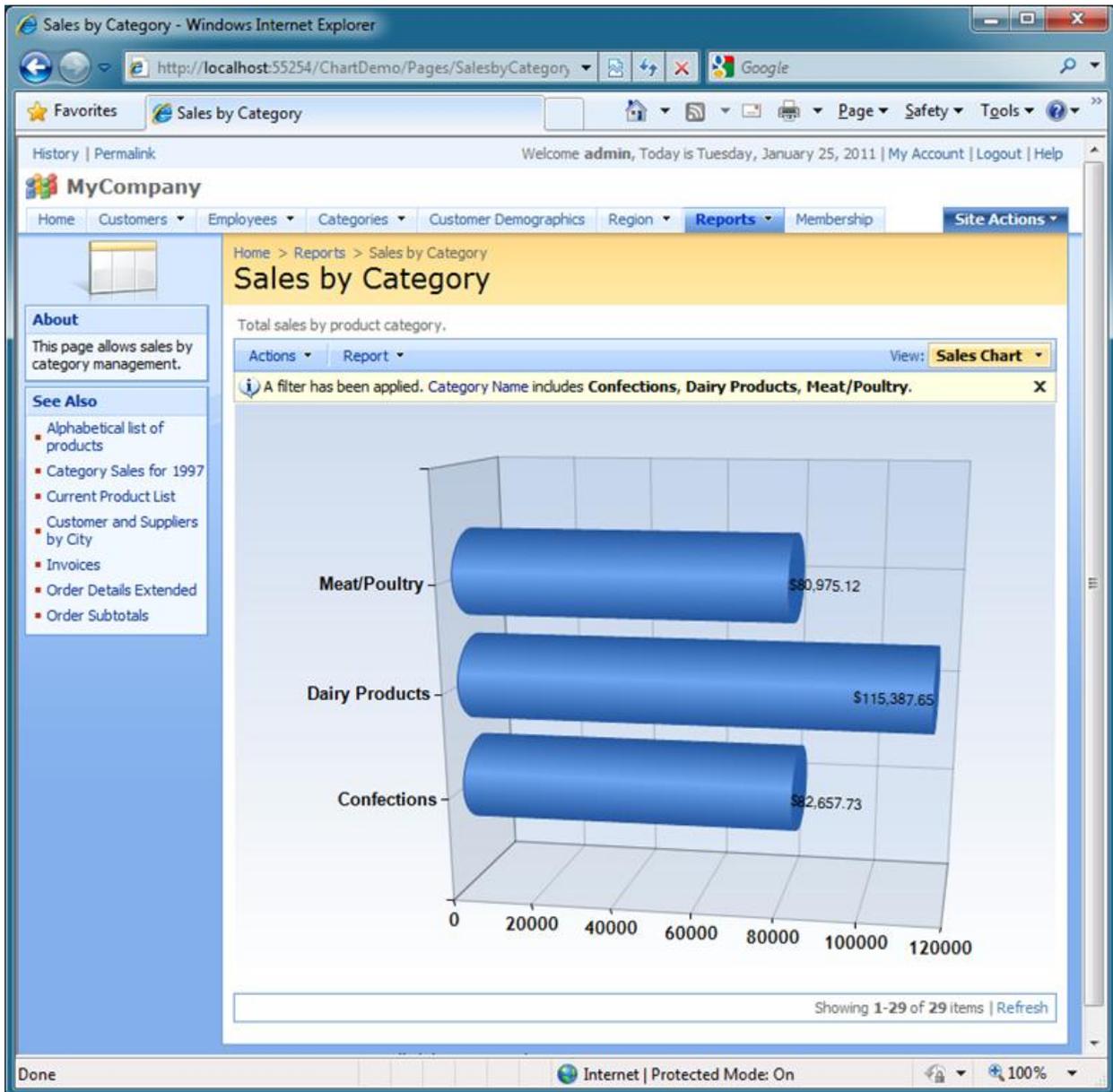
Exit the *Designer* and generate your application. Activate the same page and select *Sales Chart* option in the view selector in the right hand corner of the action bar. The following chart on the next page will be displayed.



Activate the filter in the view selector and select “Filter...” item in the popup menu of the *Category Name* option.



Select several filter options to review subset of data presented in the chart.



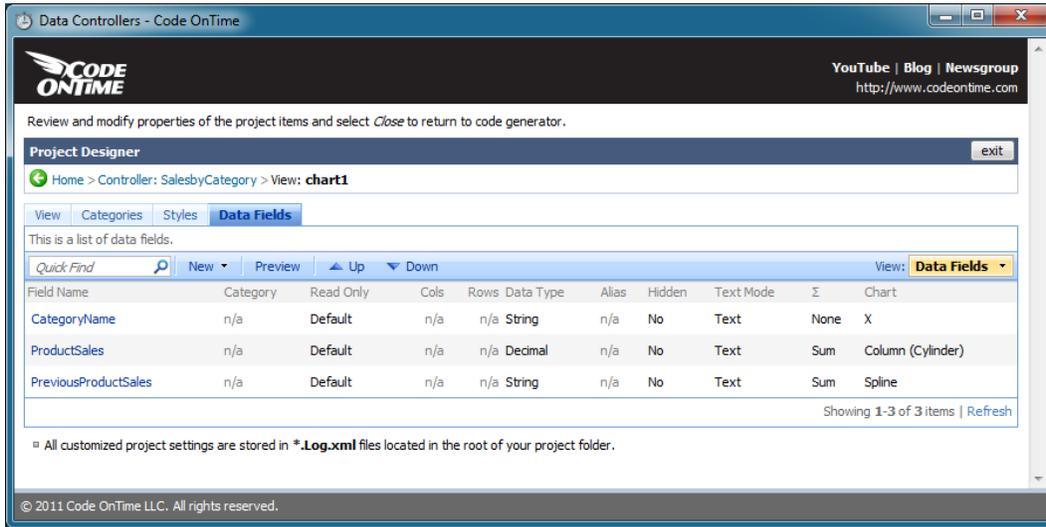
Displaying Multiple Values

The chart view is capable of displaying multiple data series. Let's add a calculated field to the same data controller to simulate the "Previous Product Sales". Select the data controller in *Designer* and activate *Fields* tab. Add a new field with name *PreviousProductSales*, indicate that the field value is calculated by SQL formula and enter the following SQL formula into *SQL Formula* text box:

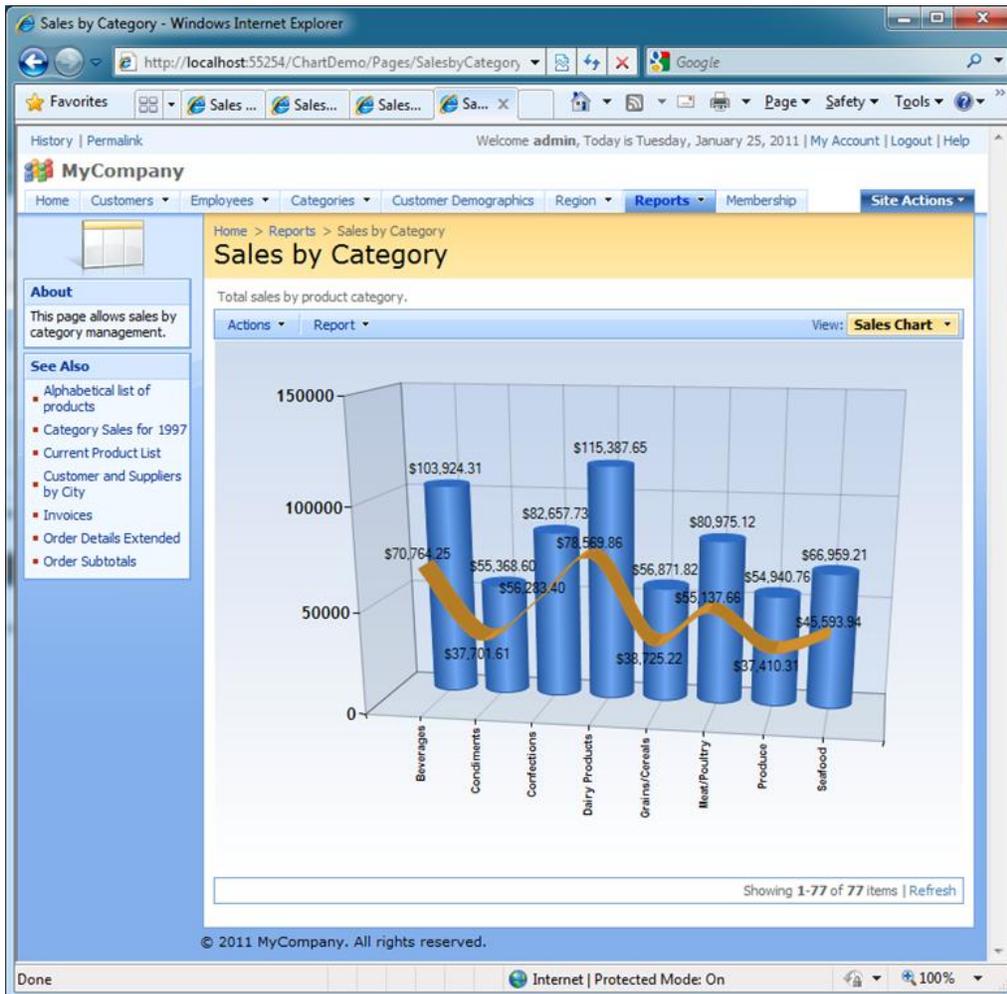
```
cast(ProductSales * Rand() as Numeric(10,2))
```

Set the label of the field to "Previous Product Sales". Set its *Data Format String* to "c" without quotes. Save the field and select *Views* tab. Select *chart1* in the list of available views. Bind the new field to

the *chart1* view and set its properties to make them look as shown in the screenshot. Notice that we are using a different *Chart* type *Column(Cylinder)* for *ProductSales*.



Run the generated application. The following chart view will be presented if you activate *Sales Chart* in the view selector. The actual spline that you will see may look different due to randomization factor of the formula that we have specified to simulate the previous sales.



Legend

You can activate a legend if you select the chart view in *Designer* and mark the check box “*Enable legend in the chart area*”. The data field header will be used as the text displayed in the chart legend.



Custom Charts

Chart views are based on the standard Microsoft Data Visualization component included with ASP.NET 4.0. Unlimited customization options are available to developers. You can quickly customize a chart view if you select “*Custom*” as *Chart*property of the data field.

All charts are generated as ASP.NET user controls stored in `~/Controls` folder of your web application. For example, the name of the chart in this sample is `~/Controls/Chart_SalesbyCategory_chart1.ascx`. The

name of a chart user control always starts with *Chart* and includes the name of the data controller and the chart view ID.

“Custom” charts are generated once only. If a “Custom” chart exists then the code generator will not make an attempt to generate the chart again. You can safely modified hundreds of the chart control properties.