

Lescasse Consulting

Getting Started with LC.PivotChart

June 4, 2015

Revised April 2, 2016

by Eric Lescasse

Lescasse Consulting
18 rue de la Belle Feuille
92100 Boulogne
France
Tel : (33) 6 47 98 13 61
eric@lescasse.com
<http://www.lescasse.com>

Table of Contents

Contents

Table of Contents	2
Requirements	3
Installation.....	3
The PivotChart workspace.....	4
The PivChart.sf User Command File	5
Basics	5
The LC.PivotChart Toolbar.....	11
The PivotGrid toolbar buttons.....	12
The Chart Toolbar Buttons	16
The Skins button.....	16
Exporting the Grid, the Chart or Both	18
Using LC.PivotChart from APL	21
Creating an instance of LC.PivotChart.....	21
Populating the LC.PivotChart Control	21
The LC.PivotChart API.....	23
The LC.PivotChart Properties	23
The LC.PivotChart Methods.....	23
Examples.....	23
Appendix 1.....	35

Introduction

The purpose of **LC.PivotChart.dll** is to provide APL+Win Users with a **PivotGrid** tool associated with a powerful **Chart** control.

LC.PivotChart is a very useful and powerful tool for analyzing and visualizing your data.

LC.PivotChart is delivered as a C# User Control which can be easily embedded in an APL+Win Form.

The **PivotGrid** and **Chart** provided by **PivotChart** can be customized through code using `□wi`, but the characteristics of this product is its very high level of interactivity which allows end users to explore their data in almost any imaginable way.

LC.PivotChart is a commercial product

LC.PivotChart requires the latest version of APL+Win.

Requirements

In order to use LC.Charts, you need:

1. To have the **.Net Framework 4.5.2+** installed on your computer
2. To have the latest version of **APL+Win** installed on your computer

Installation

WARNING: LC.PivotChart requires APL+Win v15.1+

LC.PivotChart can be downloaded from:

<http://www.lescasse.com/content/PivotChart.aspx>

(see: <http://www.lescasse.com/content/PivotChartHistory.aspx> for details about the latest version!)

where the detailed installation instructions are described.

PivotChartSetup.exe that you need to run: installation takes less than 30 seconds.

Note that there is an Uninstaller that allows you to completely uninstall LC.PivotChart from your computer if you want or need to.

The PivotChart workspace

A **PivotChart.w3** workspace is provided in APL+Win v15.1 format.

It contains 11 sample niladic APL functions:

- PivotChart
- PivotChartMom
- PivotChart2
- PivotChart3
- PivotChart4
- PivotChart5
- PivotChart6
- PivotChart7
- PivotChart8
- PivotChart9

to demonstrate how to display APL data in an APL+Win Form hosting an LC.PivotChart object.

The **PivotChartMom** is identical to PivotChart except that it uses `□mom` enums which allows for a slightly more object oriented and better documented approach. However, **PivotChartMom** is a little more verbose than **PivotChart**. You can choose either of these two ways for creating your own APL+Win forms containing a PivotChart control.

The **PivotChart2** to **PivotChart9** examples all use the **zObjects**¹ technology and the **zPivotChart** object. This is the recommended way to use LC.PivotChart.

Note that in the provided examples, the LC.PivotChart object uses the whole APL+Win Form client area, but LC.PivotChart being a User Control, you can easily place it anywhere on a form and give it the size you wish with the use of the **where**² property. The **PivotChart3** example demonstrates that.

Please also note that you can make the **PivotChart2** to **PivotChart9** examples topmost forms by entering the following expression after running those functions:

```
'ff'□wi'topmost'1
```

This way you could easily see the effect of running PivotChart API³ properties and methods from the APL Session.

Also, the PivotChart object alias name is called pc for all PivotChart example functions. If you want to test the PivotChart API from the APL Session you just need to use 'pc' as the `□wi` left argument;

Example:

¹ zObjects is a freeware developed by Lescasse Consulting

² The **where** property arguments must be expressed in pixels.

³ See a description and example of the **PivotChart API** later in this document

PivotChart2

'pc'[wi]*xlsShowLabels'1

'pc'[wi]*xChartType' 'Area'

The PivotChart.sf User Command File

Functions in the PivotChart.w3 workspace are also provided in a User Command file called PIVCHART.SF.

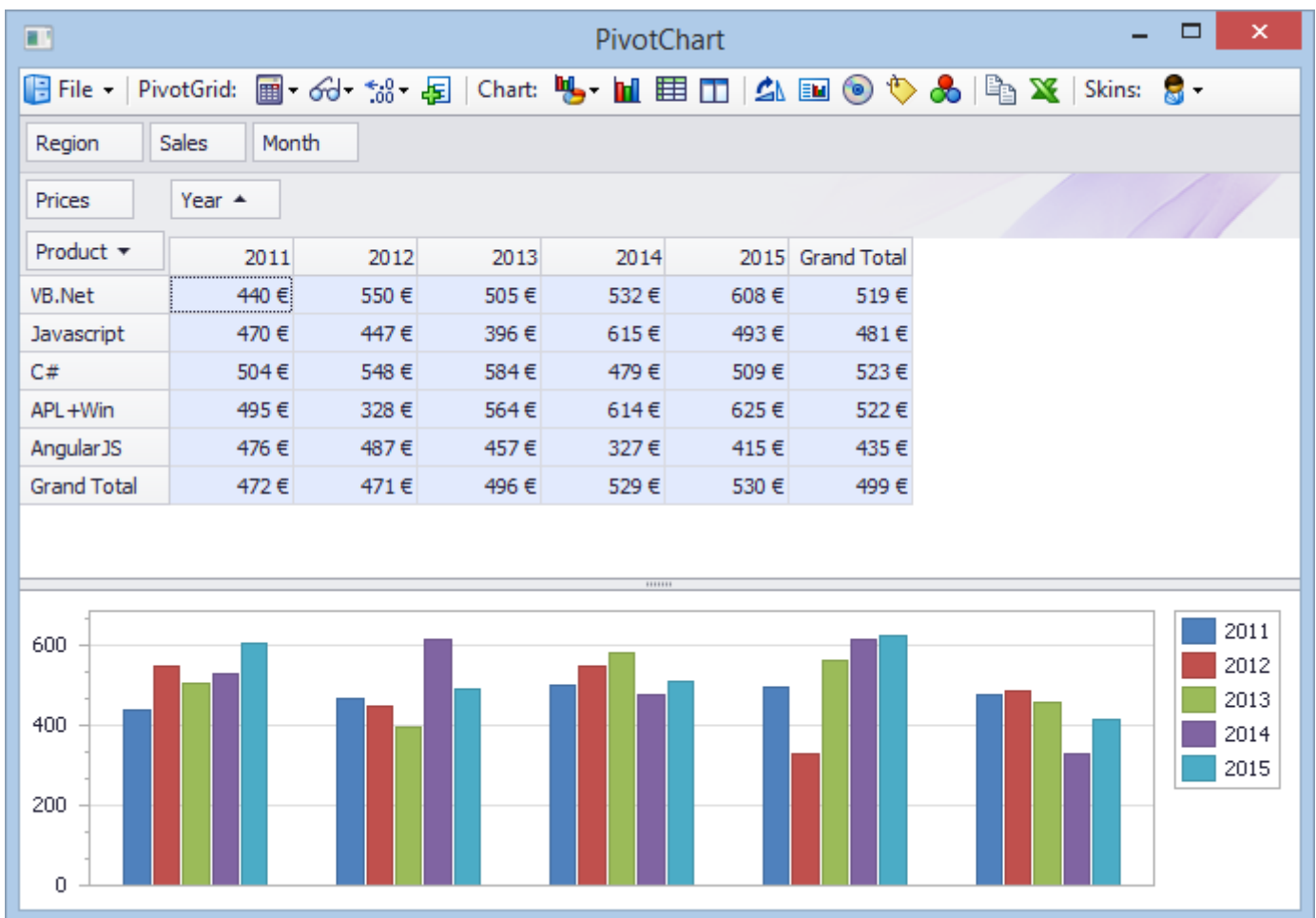
Basics

Start APL and load the **PivotChart.w3** workspace.

Run the **PivotChart** function as follows:

`PivotChart`

You should see the following APL+Win Form being displayed:

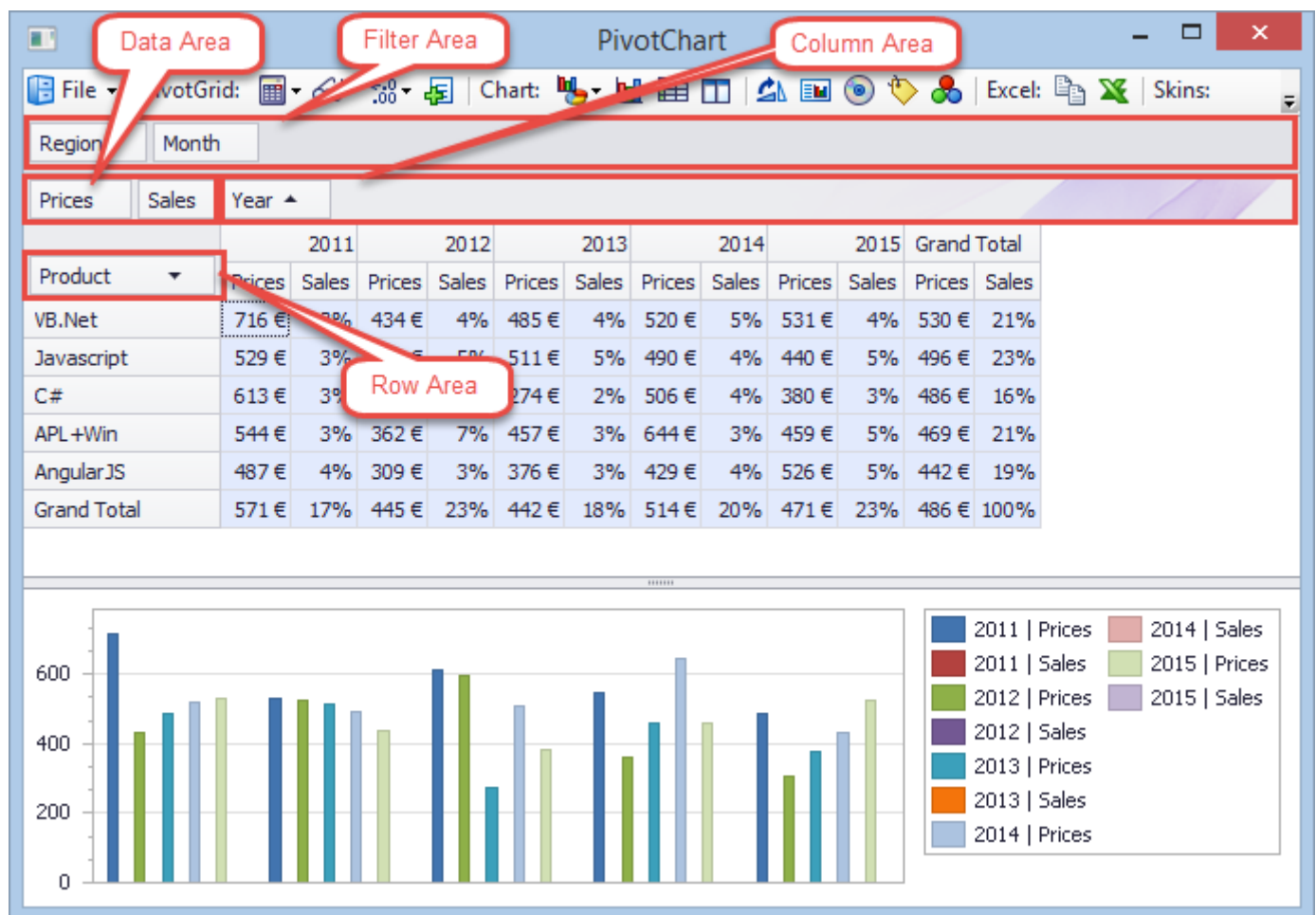


The **LC.PivotChart** User Control contains 3 parts:

- A toolbar at the top containing menus and buttons
- The **PivotGrid** in the middle
- The **Chart** at the bottom

The **PivotGrid** itself is made of various parts:

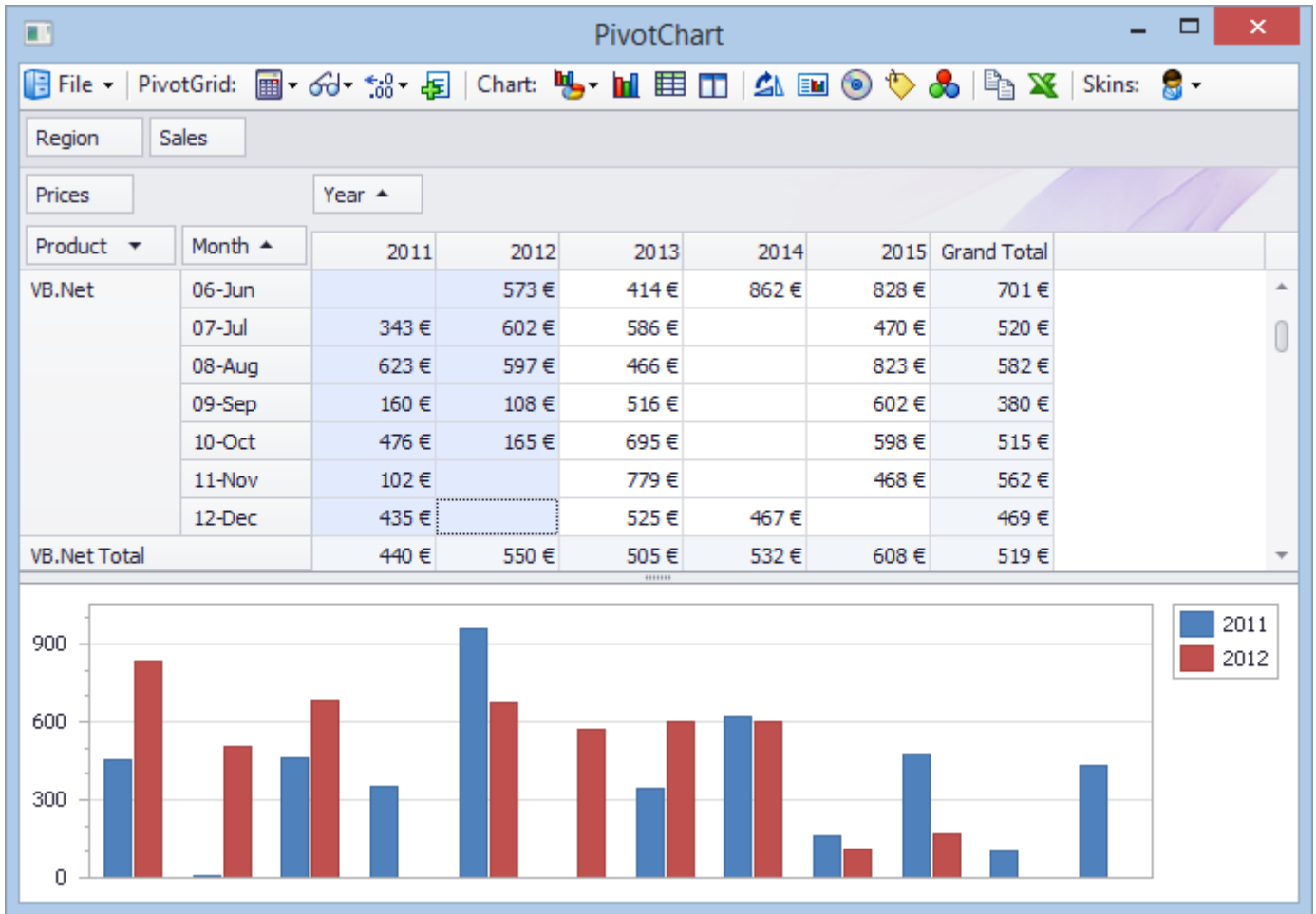
- The **Field** names which are located in 4 possible areas:
 - **row** area
 - **column** area
 - **filter** area
 - **data** area
- The **Row** Headers
- The **Column** Headers
- The **Grid** Data



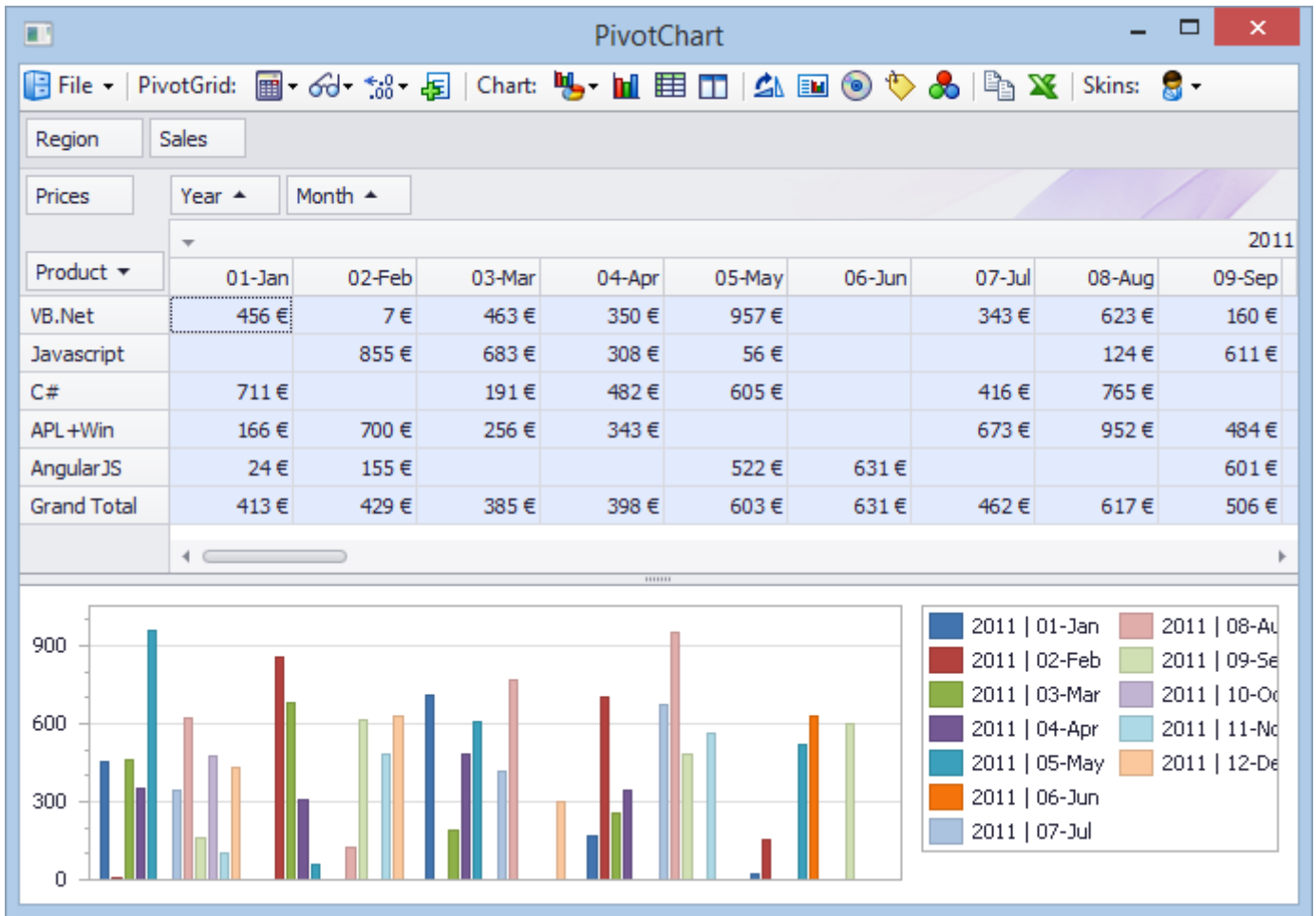
You can drag and drop Field names from one area to another area to change the layout and grouping of the resulting grid.

For example, drag and drop the **Month** field to the very right of the **Product** field name and select the first 2 columns for VB.Net only.

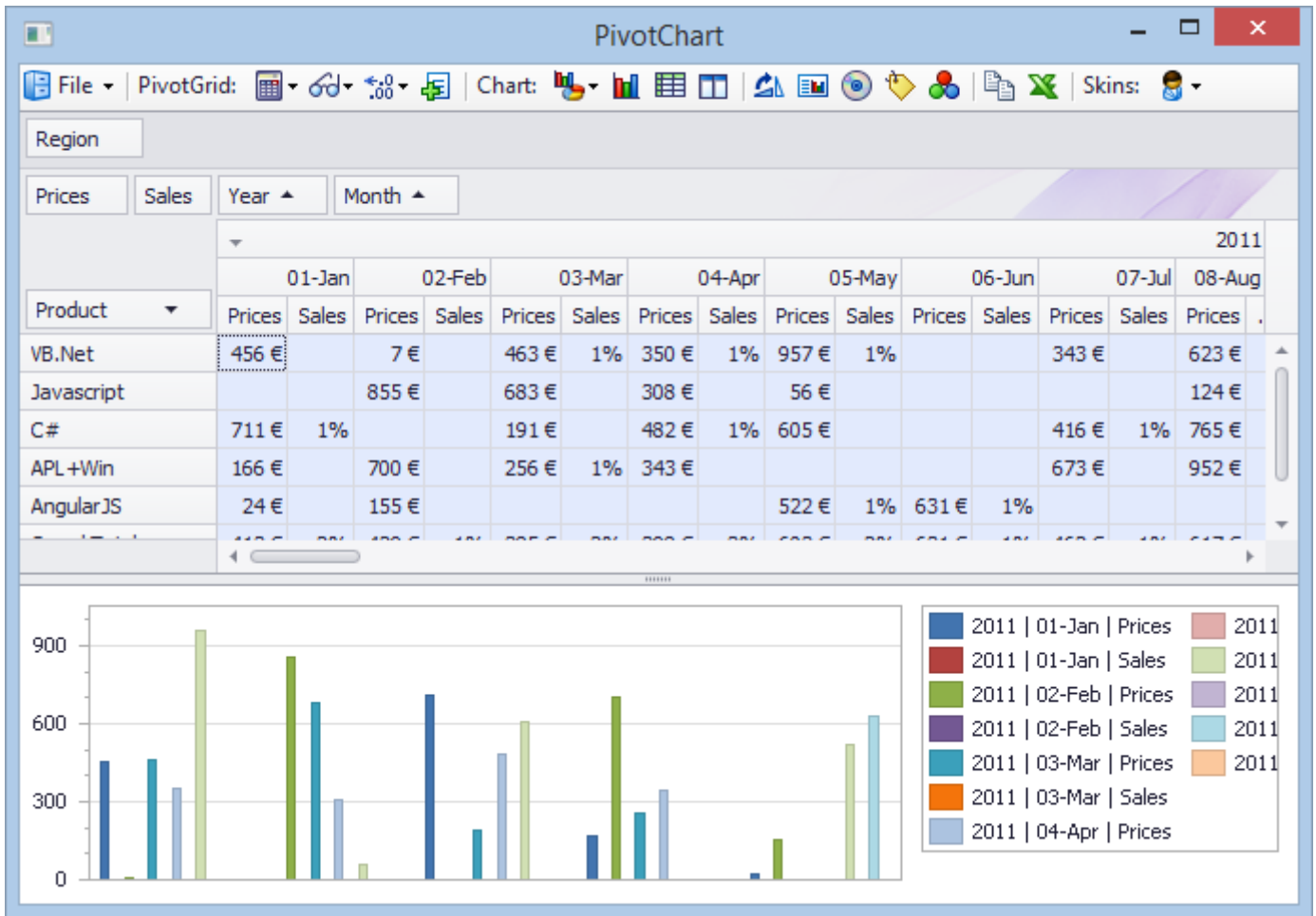
You should get something like:



Or maybe it would make more sense to display the Month field in columns: just drag the Month field and drop it to the right of the Year field; you should get:



Currently the PivotGrid displays **Prices**: you may want to also display **Sales**. Drag the **Sales** field to the immediate right of the **Prices** field: just drop it when you see the up and down arrow cursor. You should now see:



Note that each time you change the layout of the PivotGrid, all cells are selected by default and **the Chart at the bottom automatically adapts to the selected cells in the grid.**

If you hover your mouse over a cell in the grid like the VB.Net Sales for 01-Jan, a tooltip tells you what is displayed:

- Copy the Grid content to the **clipboard**
- Start **Excel** and load the PivotGrid into an Excel workbook that can then be edited and saved

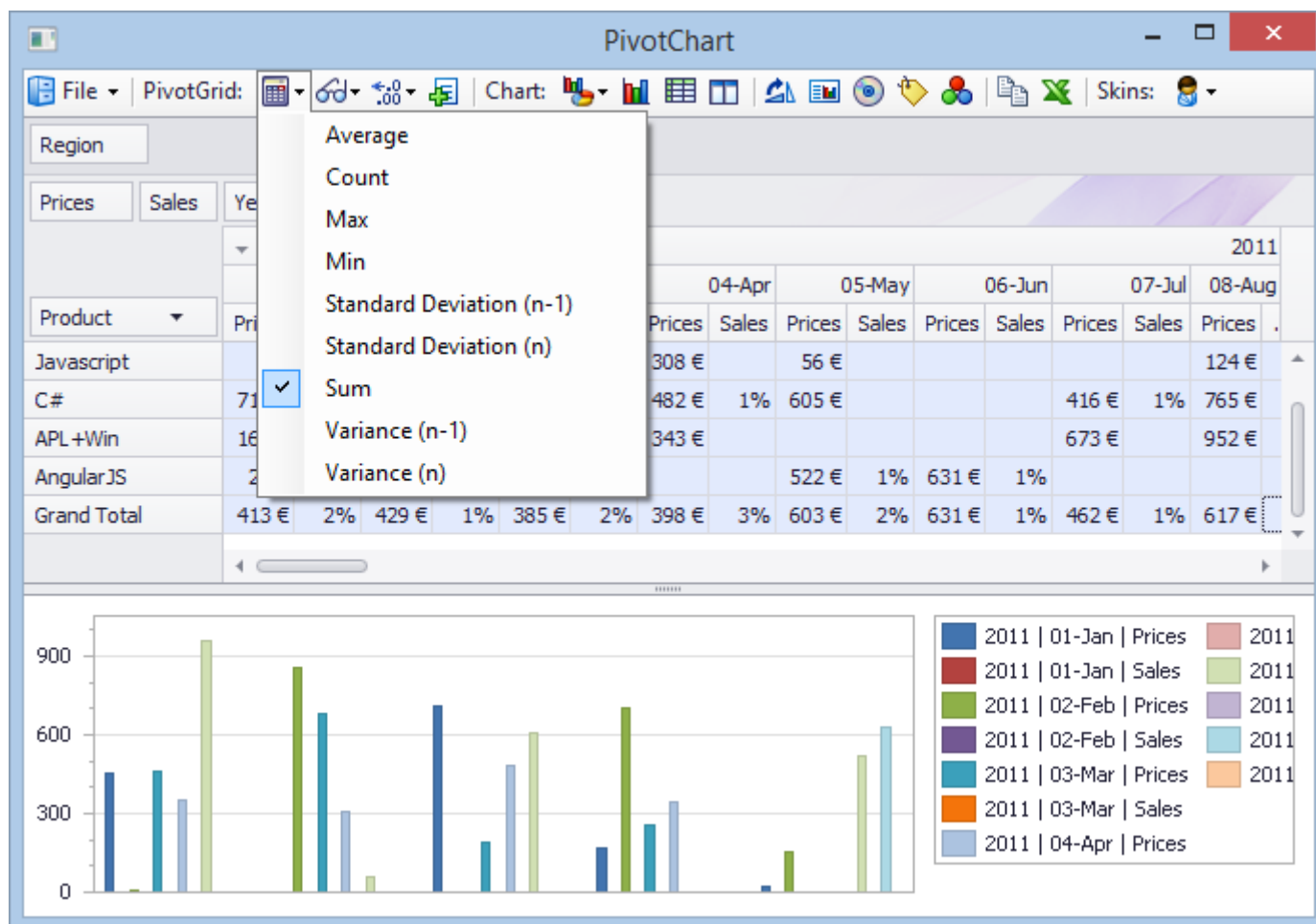
The PivotGrid toolbar buttons

Before we describe the 3 PivotGrid toolbar buttons, let's make it very clear that these buttons affect the field which is currently selected in the PivotGrid.

So, if you are currently displaying both the Prices field and the Sales field in the PivotGrid as in the last screen shot and if you want to change the number of decimals displayed in the Sales field, you must first click anywhere in the Sales field to select it. Then you can use the toolbar Decimals dropdown button to change the number of decimals for the Sales field. Your change will only affect the Sales field.

If the current PivotGrid selection spans over more than one field, there is always only one of these fields that contain the active cell. It is this field which will be affected by your toolbar button choice.

The first PivotGrid toolbar button allows you to change the **Summary Type**. It includes the following choices:

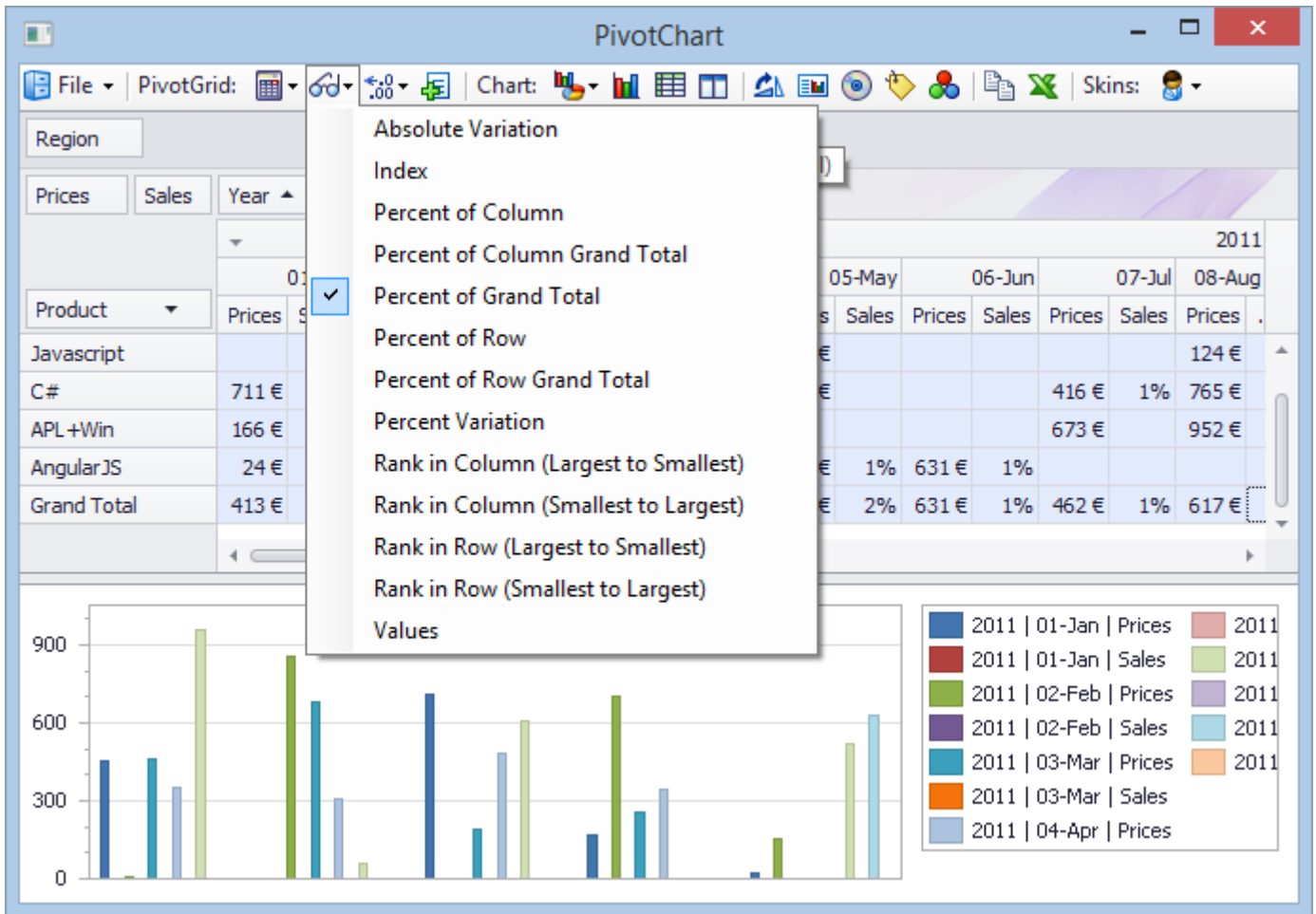


Average	the average of the values
Count	the number of values (excluding Null and DBNull values)
Custom	specifies whether calculations should be performed manually, by handling the CustomSummary event (requires custom C# programming on demand ⁴)
Max	the largest value
Min	the smallest value
Stddev	an estimate of the standard deviation of a population, where the sample is a subset of the entire population [division by n-1]
Stddevp	the standard deviation of a population, where the population is all of the data to be summarized [division by n]
Sum	the sum of the values
Var	an estimate of the variance of a population, where the sample is a subset of the entire population [division by n-1]
Varp	the variance of a population, where the population is all of the data to be summarized [division by n]

When you change the **Summary Type**, the data displayed in the grid gets immediately adjusted according to your choice and consequently the Chart also gets automatically adjusted.

Once you have chosen a Summary Type, you can use the 2nd toolbar PivotGrid button to change the **Summary Display Type**. You can choose among:

⁴ Please contact Lescasse Consulting (eric@lescasse.com) for such a need.



Absolute Variation	the absolute variation between the current value and the previously calculated value for the current field is displayed in the cell
Default	the calculated summary values are displayed "as is" (this is the same as Values)
Index	an index value that reflects the summary value's importance in its row and column context. This value is calculated as follows: $((\text{value in cell}) \times (\text{Grand Total of Grand Totals})) / ((\text{Grand Row Total}) \times (\text{Grand Column Total}))$
Percent of Column	for data cells, the percentage of the column's total value is displayed; for total cells, the percentage of the column's grand total value is displayed
Percent of Column Grand Total	percent of the summary in the Column Grand Total value
Percent of Grand Total	percent of the summary in the Grand Total value
Percent of Row	for data cells, the percentage of the row's total value is displayed; for total cells, the percentage of the row's grand total value is displayed

Percent of Row Grand Total	percent of the summary in the Row Grand Total value
Percent Variation	the percentage variance between the current value and the previously calculated value for the current field is displayed in the cell
Rank in Column (Largest to Smallest)	the summary value's rank in its column, with the largest value in the column as 1, and each smaller value assigned a higher rank
Rank in Column (Smallest to Largest)	the summary value's rank in its column, with the smallest value in the column as 1, and each larger value assigned a higher rank
Rank in Row (Largest to Smallest)	the summary value's rank in its row, with the largest value in the row as 1, and each smaller value assigned a higher rank
Rank in Row (Smallest to Largest)	the summary value's rank in its row, with the smallest value in the row as 1, and each larger value assigned a higher rank
Values	the summary values

So, the data displayed in the Grid depends of both the **Summary Type** and the **Summary Display Type**.

The 3rd toolbar PivotGrid button allows you to change the number of decimals for the field containing the active cell.

When you select a number of decimals, and if you have defined a format string for the active field, the system makes it best attempt to respect your format string except that it uses regular expressions to change the number of decimals according to your choice.

So, for example, if you have defined the format string to be:

```
"#.0 €;-#.0 €;"
```

And then change the number of decimals to 2, the system now displays the data using the following format string:

```
"#.00 €;-#.00 €;"
```

Note that selecting the **Count** Summary Type forces the format string to temporarily display raw numbers with no decimals, no currency symbol, i.e. numbers are displayed as plain integers.

Similarly, selecting any of the **Rank** Summary Display Types forces the format string to temporarily display raw numbers with no decimals, no currency symbol, i.e. numbers are displayed as plain integers.

Also, selecting any of the **Percent** Summary Display Types forces the format string to temporarily display percentages, using the Percentage Format String you've defined in the layout variable, or using the currently selected number of decimals if you have not defined any Percentage Format String in the layout variable.










However, the Format String you have defined in the layout variable, if any, is never lost and if you choose a Summary Type and Summary Display Type which do not force specific format string, it is the one which you have defined in the layout variable which will be used.

The 4th PivotGrid toolbar button allows you to Show or Hide totals in the grid.

The Chart Toolbar Buttons

The Chart toolbar buttons let you customize the way the Chart associated with the Grid is displayed.

First let's emphasize that some of the toolbar buttons do not always have an effect on the chart, depending on the chart type. For example, the 9th Chart toolbar button which displays or hides point symbols, does not have any effect on bar charts (simply because point symbols do not make sense for a bar chart).

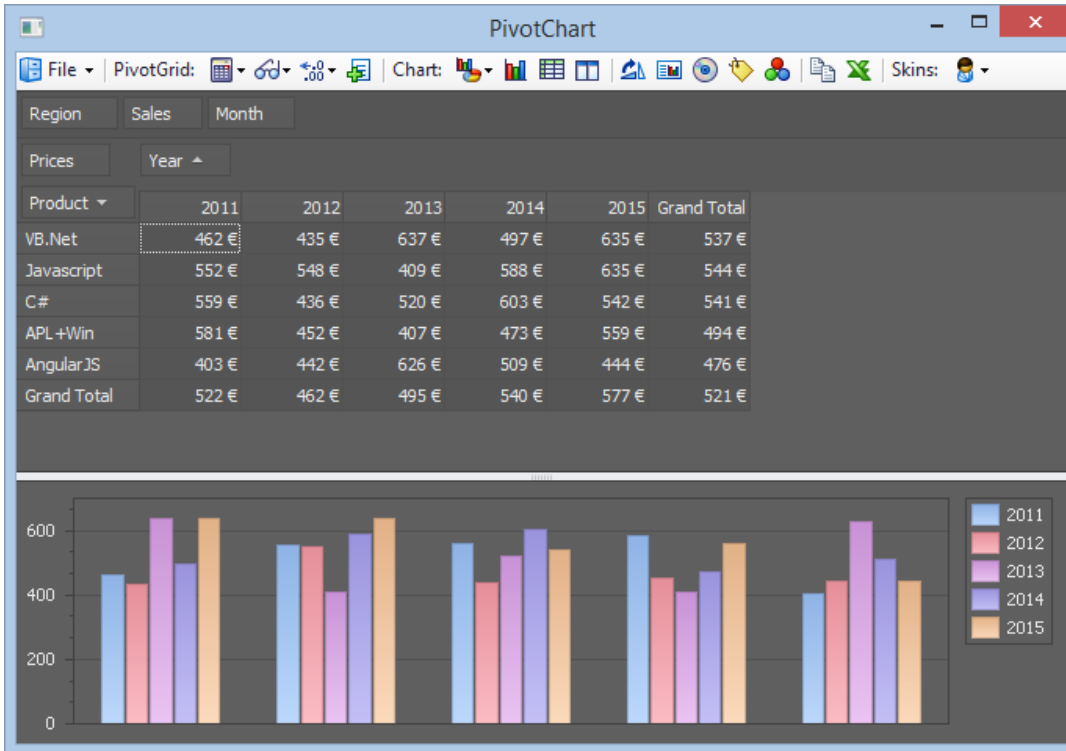
	The 1 st Chart toolbar button lets you choose the type of chart you want to display. There are about 40 possible choices including 2D and 3D charts.
	The 2 nd Chart toolbar button is a toggle button which lets you maximize or un-maximize the chart.
	The 3 rd Chart toolbar button is a toggle button which lets you maximize or un-maximize the grid.
	The 4 th Chart toolbar button is a toggle button which lets you display the chart to the right of the grid or below the grid.
	The 5 th Chart toolbar button is a toggle button which allows you to rotate the chart 90 degrees.
	The 6 th Chart toolbar button is a toggle button which lets you hide or show the chart legend.
	The 7 th Chart toolbar button allows you to change the legend position. There are 24 possible legend positions (inside or outside the chart) and each click on the 7 th chart toolbar button moves the legend to the next position. Note that you can Shift+Click the 7 th toolbar button to return to the previous position(s).
	The 8 th Chart toolbar button is a toggle button allowing you to display or hide labels on the chart. Labels display point values and in this version of PivotChart, the label values are displayed with 2 decimals regardless of the corresponding field format string.
	The 9 th Chart toolbar button is a toggle button letting you display or hide point symbols.

The Skins button

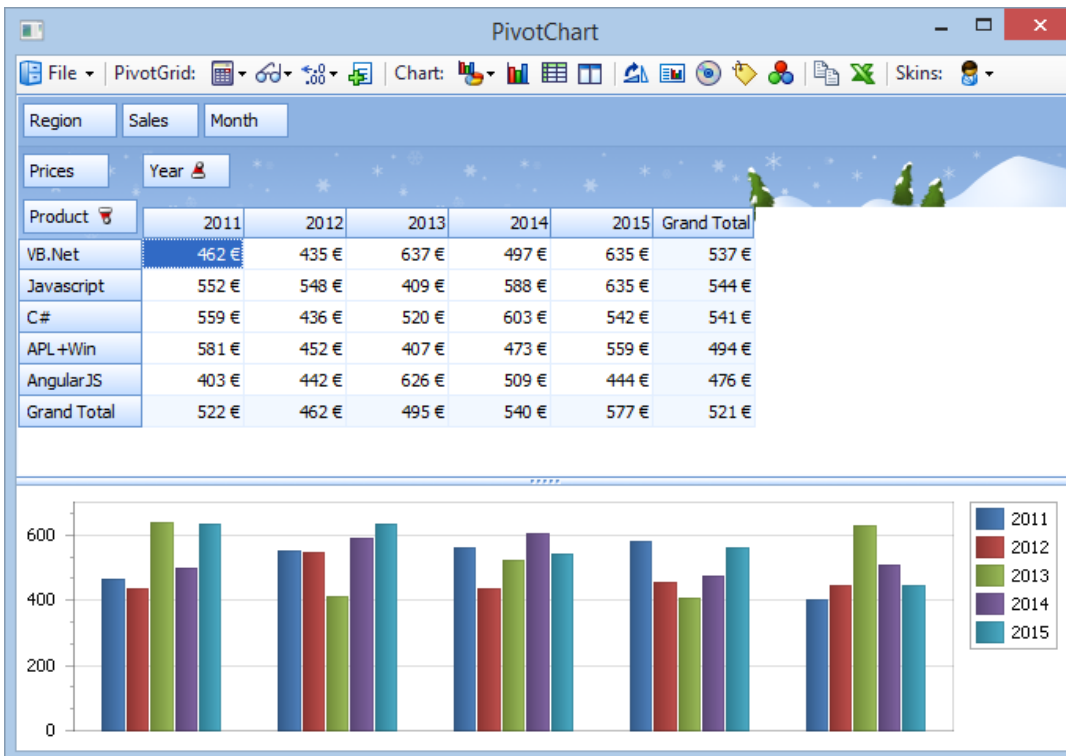
The Skins dropdown button lets you choose among 24 possible different skins for the PivotChart control.

Note that, in this version of LC.PivotChart, choosing a Skin has no effect on the PivotChart toolbar.

Let's for example choose the DevExpress Dark Style skin:



Or the Xmas 2008 Blue skin (which you can automatically programmatically select in your application when approaching christmas time):



Exporting the Grid, the Chart or Both

The File menu has many options for exporting the Grid, the Chart or both the Grid and the Chart to many different formats:

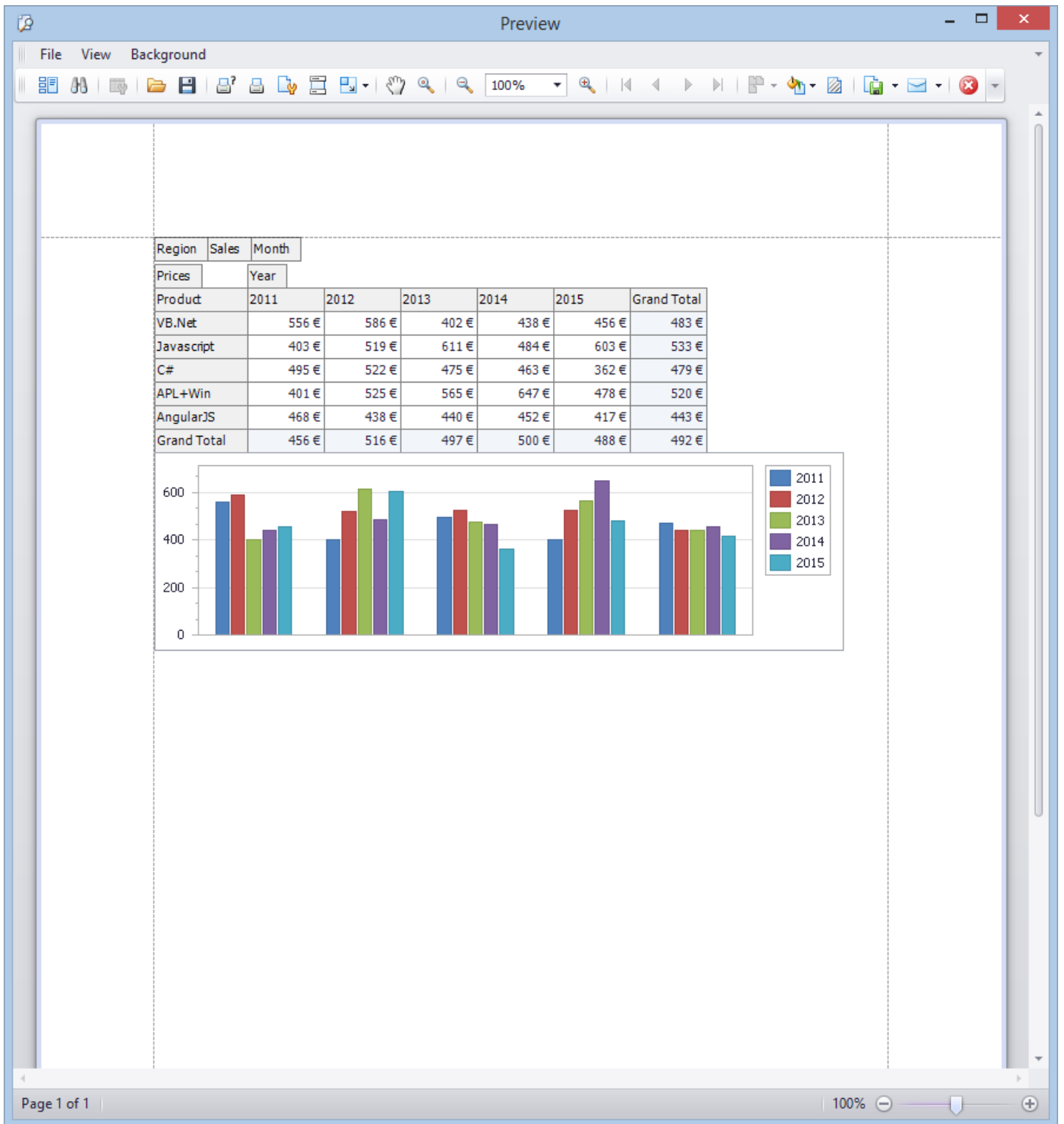
The screenshot shows the PivotChart application window. The File menu is open, displaying various export options. The 'Export Chart to Image' option is selected, and a sub-menu is visible showing image formats: BMP, EMF, GIF, JPG, PNG, TIFF, and WMF. The background shows a data grid and a bar chart.

	2013	2014	2015	Grand Total
	4.77%	5.41%	5.82%	23.31%
	1.66%	3.76%	3.66%	15.96%
	37%	2.37%	18.14%	
	28%	4.38%	23.98%	
	86%	3.77%	18.61%	
	69%	20.00%	100.00%	

The bar chart displays data for five categories: VB.Net, Javascript, C#, APL+Win, and AngularJS. The Y-axis ranges from 0 to 0.05. The legend indicates the years: 2011 (blue), 2012 (red), 2013 (green), 2014 (purple), and 2015 (teal).

If you choose one of the first 3 options, you will be presented with an interactive Print Preview form.

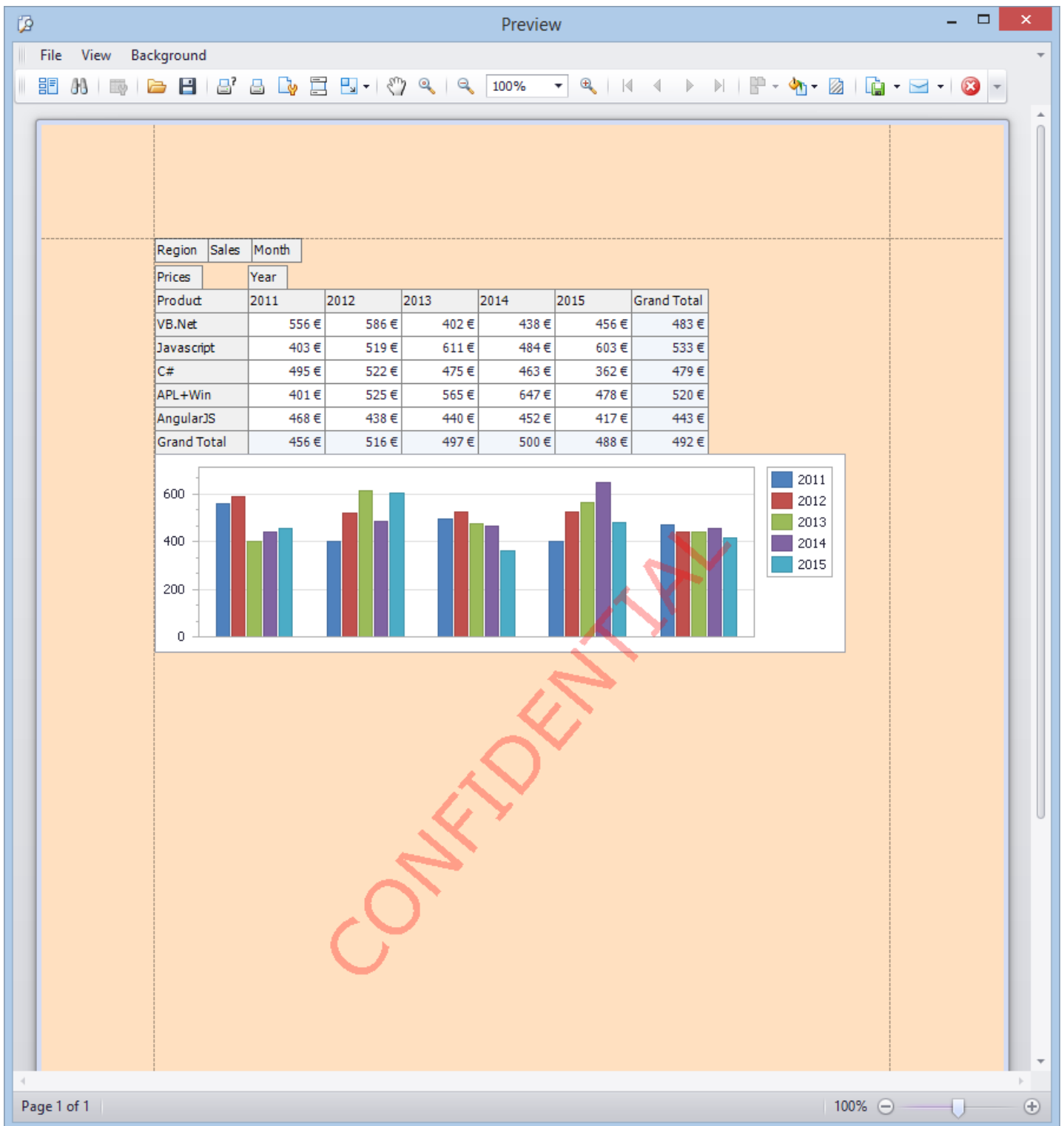
For example, assuming you choose **Export Both with Preview**, the following form is displayed:



You can then :

- drag and drop the margins to adjust them with your mouse
- use the File menu to change the Page Steup, Print, Export the document or automatically send it by Email

You can also use the Background menu to change the background color and/or add a Watermark:



Using LC.PivotChart from APL

Creating an instance of LC.PivotChart

LC.PivotChart is a C# ActiveX User Control: as such, it requires a form to host it.

So the first step is to create an APL+Win Form:

```
'ff'⎕wi '*Create' 'Form' ('*scale'5)
```

You can then create an instance of LC.PivotChart.PivotChart as a child of this form:

```
'ff'⎕wi '*.pc.Create' 'LC.PivotChart.PivotChart'
```

You'll generally want to force this control to use the entire form area, but are not forced to and you can use the where property to locate it and size it as you want.

```
'ff'⎕wi '*.pc. where' -1 -1,2+'ff'⎕wi '*size'
```

Most often you'll also want to handle the APL+Win Form **onResize** event to automatically resize the LC.PivotChart control when the form is resized⁵.

Populating the LC.PivotChart Control

The next step is to populate the **LC.PivotChart** control with your own data.

This is easily done by creating a "layout" variable and passing it to the **XPopulate** method.

⁵ Look at the **PivotChart** APL function in the **PivotChart.w3** workspace (and also reproduced in **Appendix 1**) for an example.

The layout variable is a nested matrix currently⁶ using 9 columns defined as follows:

<code>layout[;1]</code>	field name
<code>layout[;2]</code>	field data (vector)
<code>layout[;3]</code>	field area (filter, row, col or data)
<code>layout[;4]</code>	field summary type (average, count, custom, max, min, stddev, stddevp, sum, var, varp or "" if no summary type applies)
<code>layout[;5]</code>	field summary display type (absvar, default or "", index, %col, %colgt, %gt, %row, %rowgt, %var, rankcoll2s, rankcols2l, rankrowl2s, rankrows2l, value)
<code>layout[;6]</code>	column width (or -1 to use a best fit column width)
<code>layout[;7]</code>	percent format string ⁷ (format string to be used when displaying percentages or "" to let the system choose the percent format string)
<code>layout[;8]</code>	format string (format string to be used when displaying values or "" to let the system choose the format string)
<code>layout[;9]</code>	field sort order (ascending, descending or "")

Once you have created your layout variable, just use the following instruction to populate the LC.PivotChart object:

```
'ff'[wi]*.pc.XPopulate'layout
```

From there, you're in business and can start exploring your data in all kinds of ways using the LC.PivotChart control.

⁶ Note that more columns could be added in later releases

⁷ All format strings must be expressed as valid Microsoft.Net format strings (see : TODO: complete here)

The LC.PivotChart API

Very often what is described above will be enough programming effort for using LC.PivotChart or letting your own end users use LC.PivotChart.

You can rely on them to interactively use the Control to explore the data by themselves.

However, LC.PivotChart includes its own API and you can programmatically control it from APL using `wi`.

The LC.PivotChart Properties

LC.PivotChart includes the following properties:

Properties

<code>xAreaTypes</code>	<code>xIsGridMaximized</code>	<code>xSkin</code>
<code>xChartType</code>	<code>xIsShowLabels</code>	<code>xSkins</code>
<code>xChartTypes</code>	<code>xIsShowLegend</code>	<code>xSummaryDisplayTypes</code>
<code>xEnableColumnFit</code>	<code>xIsShowMarkers</code>	<code>xSummaryTypes</code>
<code>xFieldNames</code>	<code>xIsShowTotals</code>	<code>xToolbarVisible</code>
<code>xIsChartMaximized</code>	<code>xIsTileHorizontal</code>	
<code>xIsChartRotated</code>	<code>xMaxAllowedSeries</code>	

The LC.PivotChart Methods

LC.PivotChart includes the following methods:

Methods

<code>XCopyToClipboard</code>	<code>XResumeLayout</code>	<code>XSetPctFormatString</code>
<code>XGetFormatString</code>	<code>XSelectAllCells</code>	<code>XSetSummaryDisplayType</code>
<code>XGetPctFormatString</code>	<code>XSetArea</code>	<code>XSetSummaryType</code>
<code>XGetSummaryDisplayType</code>	<code>XSetColumnWidth</code>	<code>XSuspendLayout</code>
<code>XGetSummaryType</code>	<code>XSetDecimals</code>	
<code>XPopulate</code>	<code>XSetFormatString</code>	

Examples

Using the LC.PivotChart API is straightforward.

Load the **PivotChart.w3** delivered workspace

Run the **PivotChart2** function:

```
PivotChart2
```

Note that zObjects use alias objects names and that the PivotChart object is called **pc** in function **PivotChart2**.

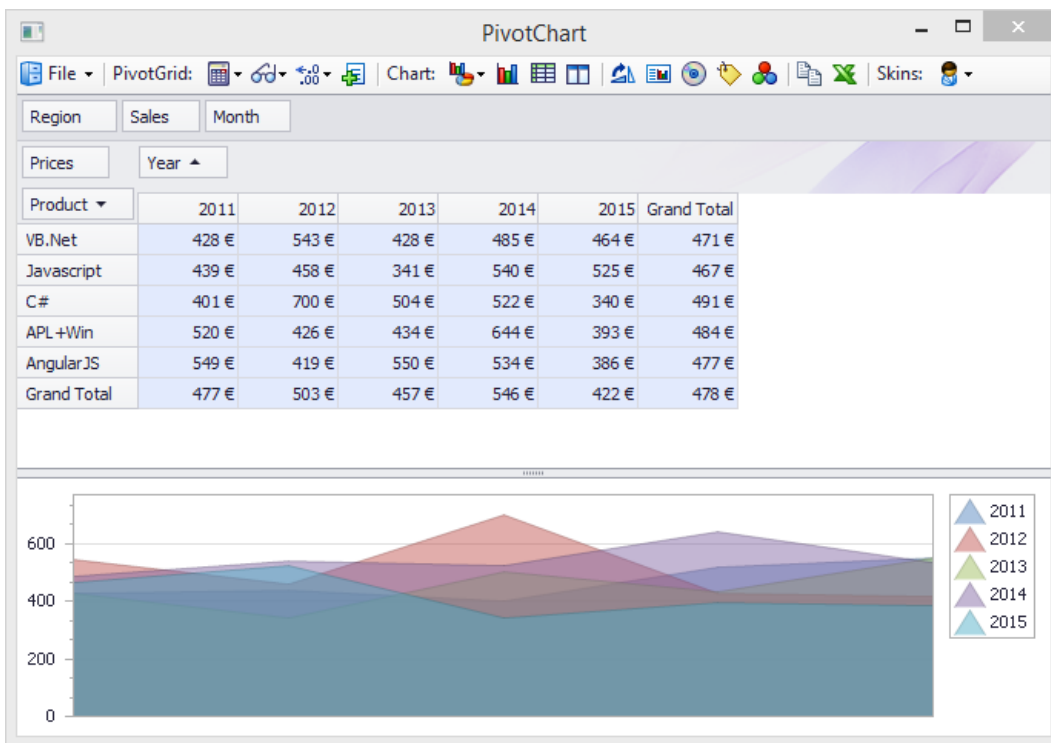
Then, you can try the following expressions and watch the effect on the LC.PivotChart control:

```

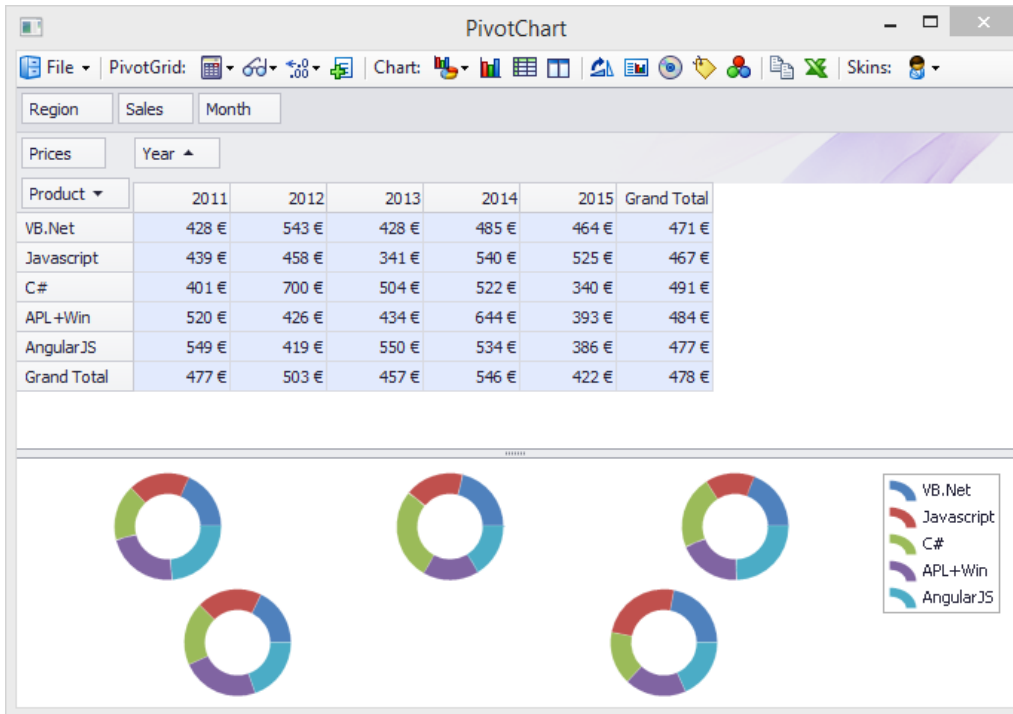
80 TELPRINT =>[2]'pc'[]wi'*xChartTypes '
area3d          points          spline
area           radararea       stackedarea3d
bubble        radarline       stackedarea
candlestick   radarpoint      stackedbar3d
doughnut3d    sidebysidebar3d  stackedbar
doughnut      sidebysidebar    stackedline3d
fullstackedbar3d sidebysidefullstackedbar3d stackedline
fullstackedbar sidebysidefullstackedbar stackedsplinearea3d
line3d        sidebysidestackedbar3d stackedsplinearea
line          sidebysidestackedbar steparea3d
manhattanbar spline3d         steparea
pie3d         splinearea3d    stepline3d
pie           splinearea     stepline

```

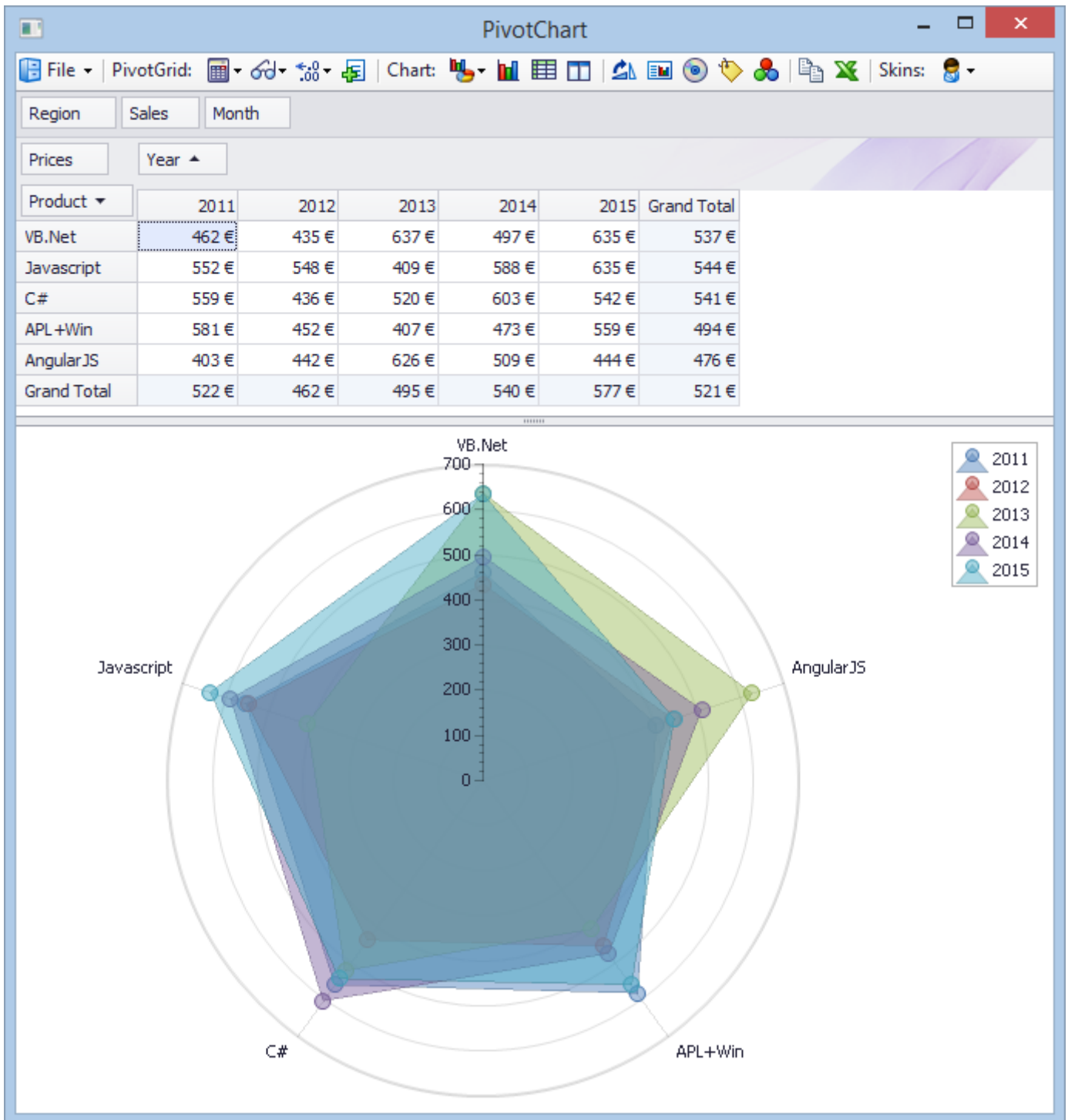
```
'pc'[]wi'*xChartType' 'area'
```



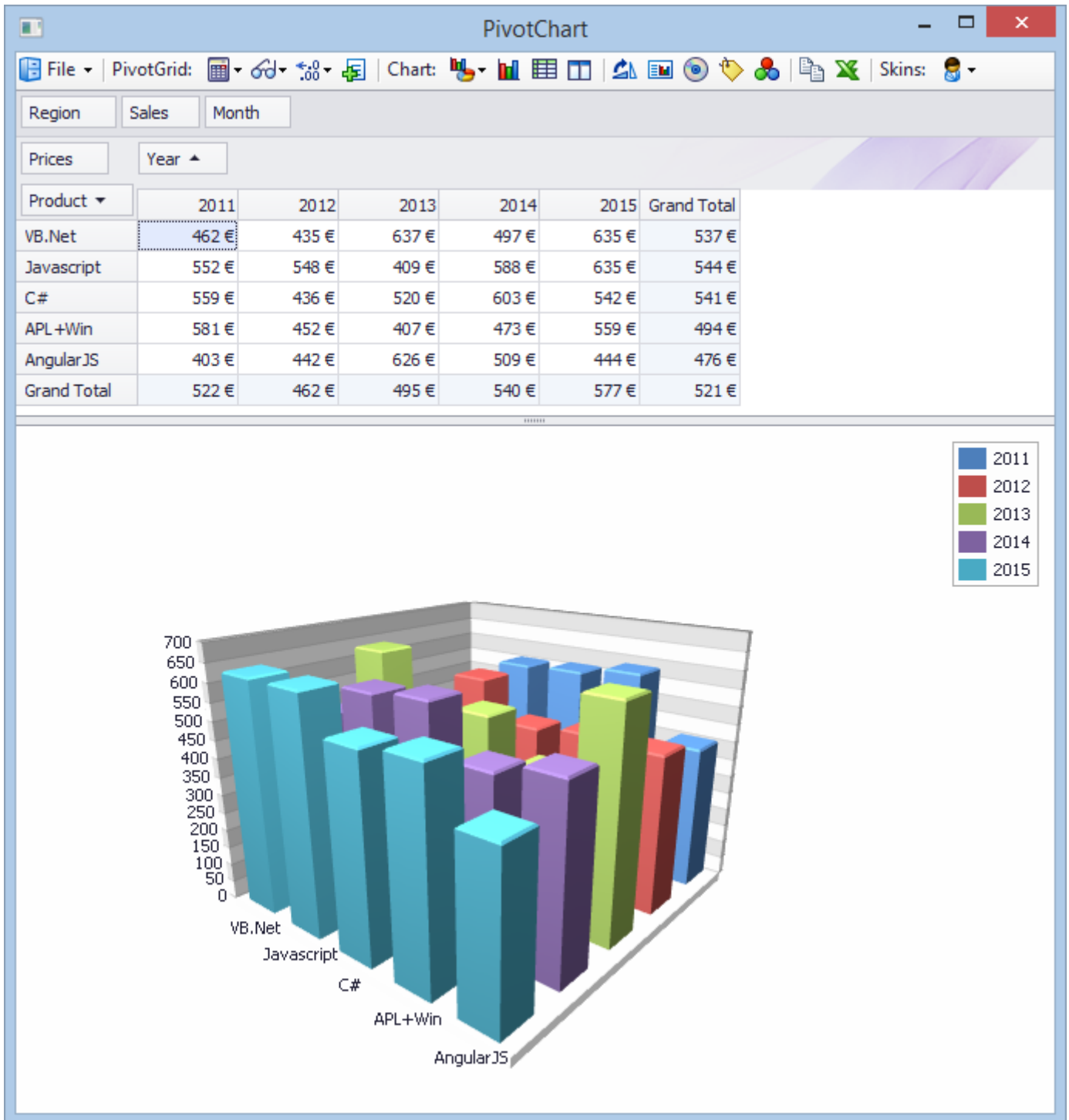
'pc'wi'*xChartType' 'doughnut'



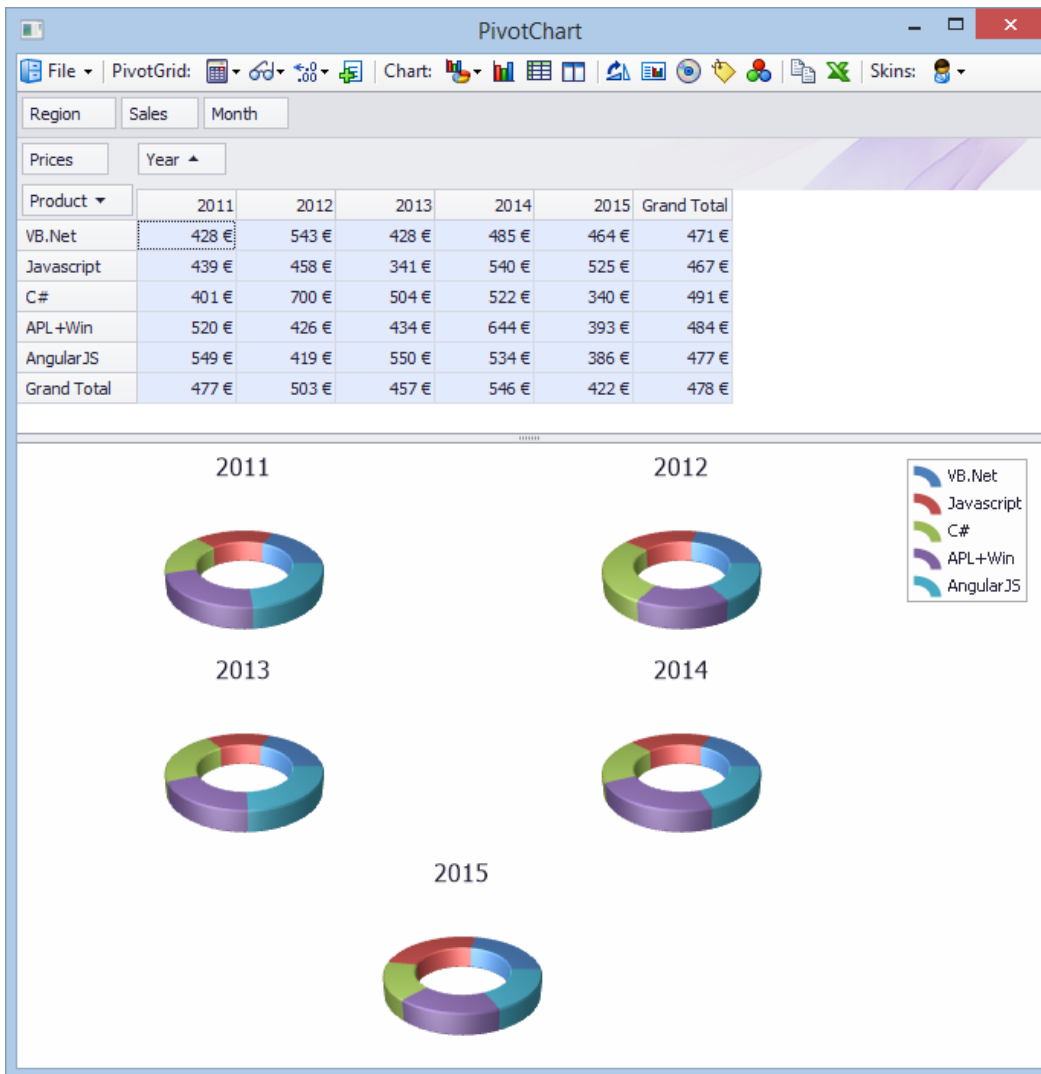
'pc'[]wi'*xChartType' 'radararea'



'pc' □wi'*xChartType' 'manhattanbar'

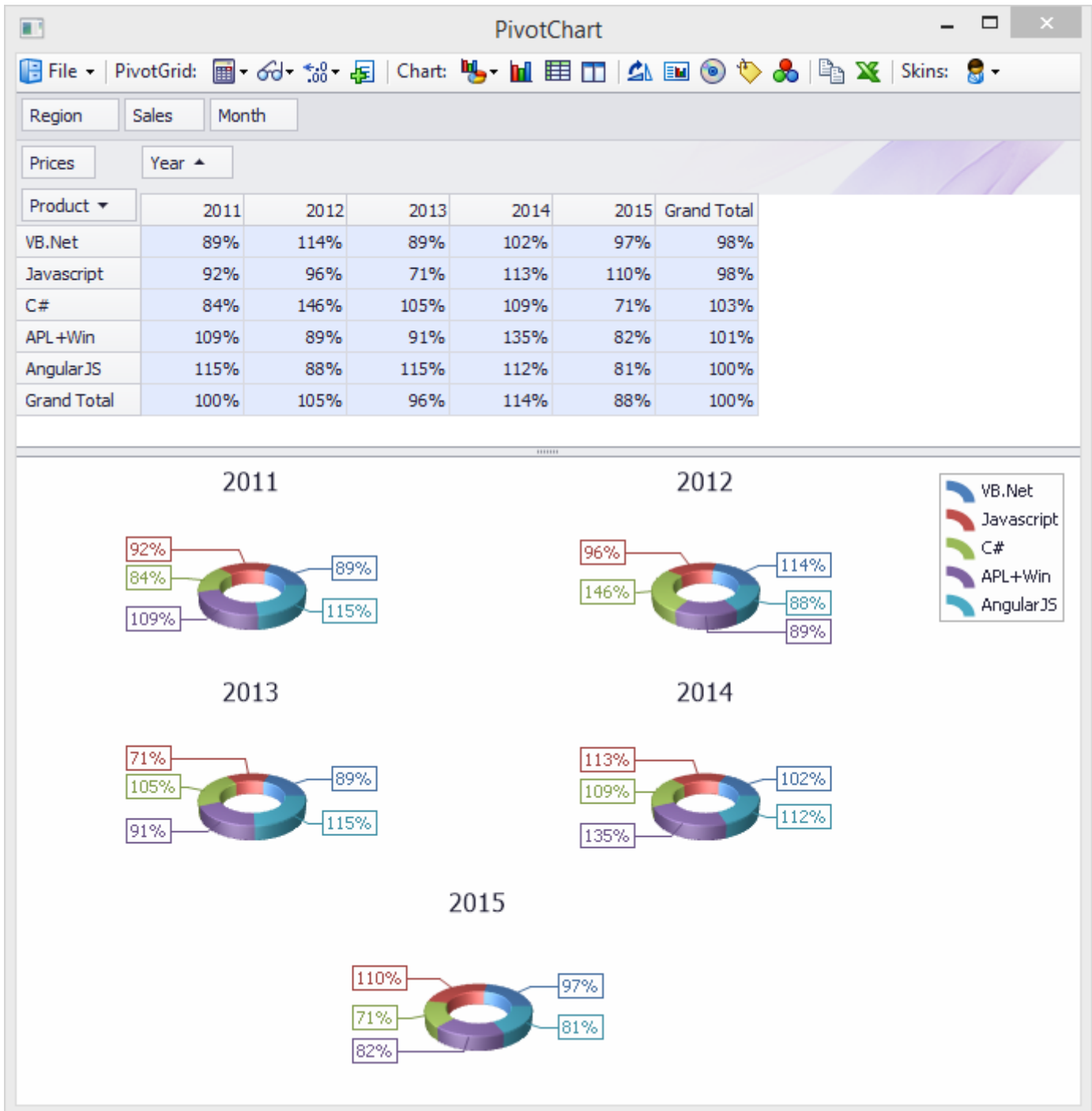


'pc'wi'*xChartType' 'doughnut3d'



```
80 TELPRINT >[2] 'pc'wi'*xSummaryDisplayTypes'
absvar  index  %colgt  %row  %var      rankcols21  rankrows21
default %col  %gt     %rowgt rankcol12s  rankrow12s
```

```
'pc'\wi'*XSetSummaryDisplayType' 'prices' '%gt'
'pc'\wi'*xIsShowLabels'1
```

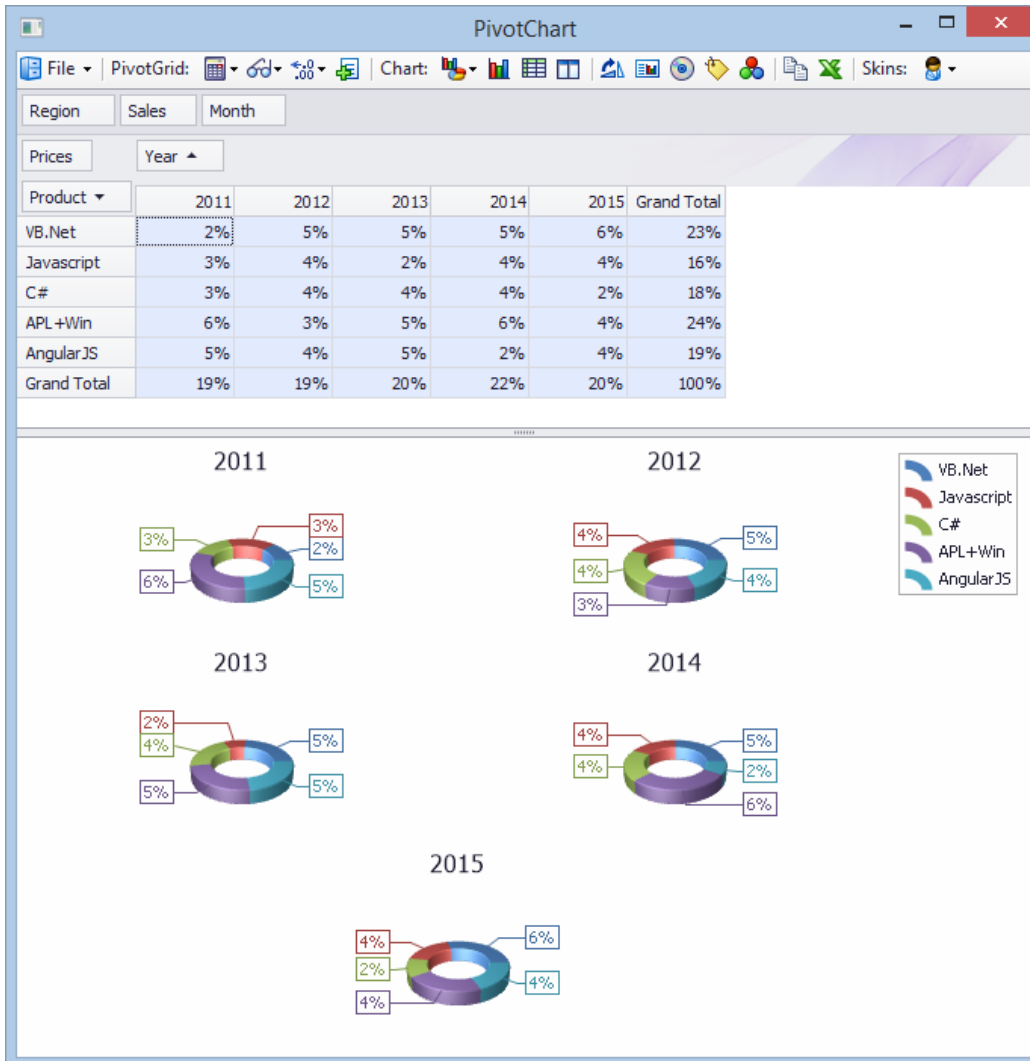


Remember that the Summary Display Type, as well as the Summary Type and the Number of Decimals are field dependent, so we must supply the field name (not case sensitive).

Note that, even though we chose %gt, i.e. percent of grand total, the displayed percentages are around 100% in each cell: this may surprise you at first, but always remember that the displayed values also depend on the Summary Type. In the current case, the Summary Type is set to Average, so the result makes sense.

Most often, when displaying one of the “percent” Summary Display Types, we want to choose a Summary Type of **Sum** or **Count**.

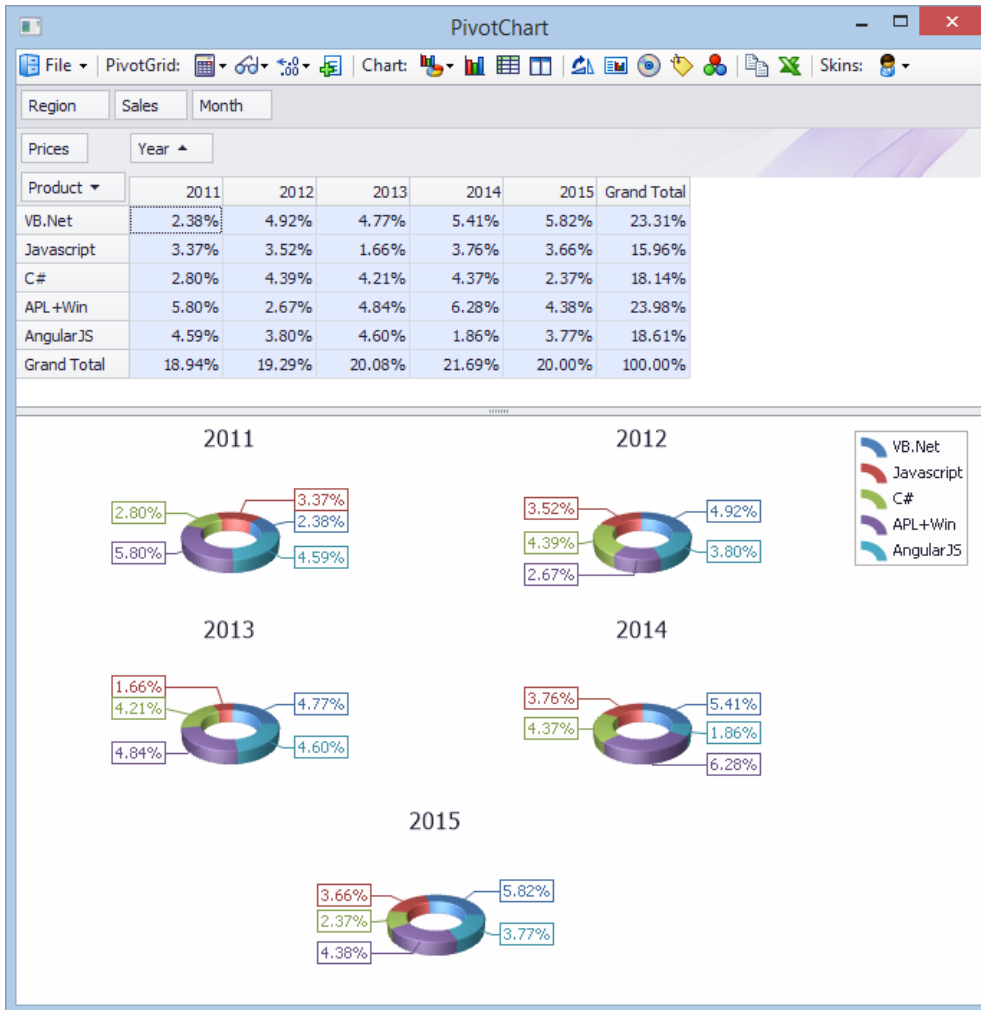
```
'pc'[]wi '*XSetSummaryType' 'prices' 'sum'
```



The results now makes more sense.

Let’s change the number of decimals to 2:

```
'pc'wi '*XSetDecimals' 'prices' 2
```



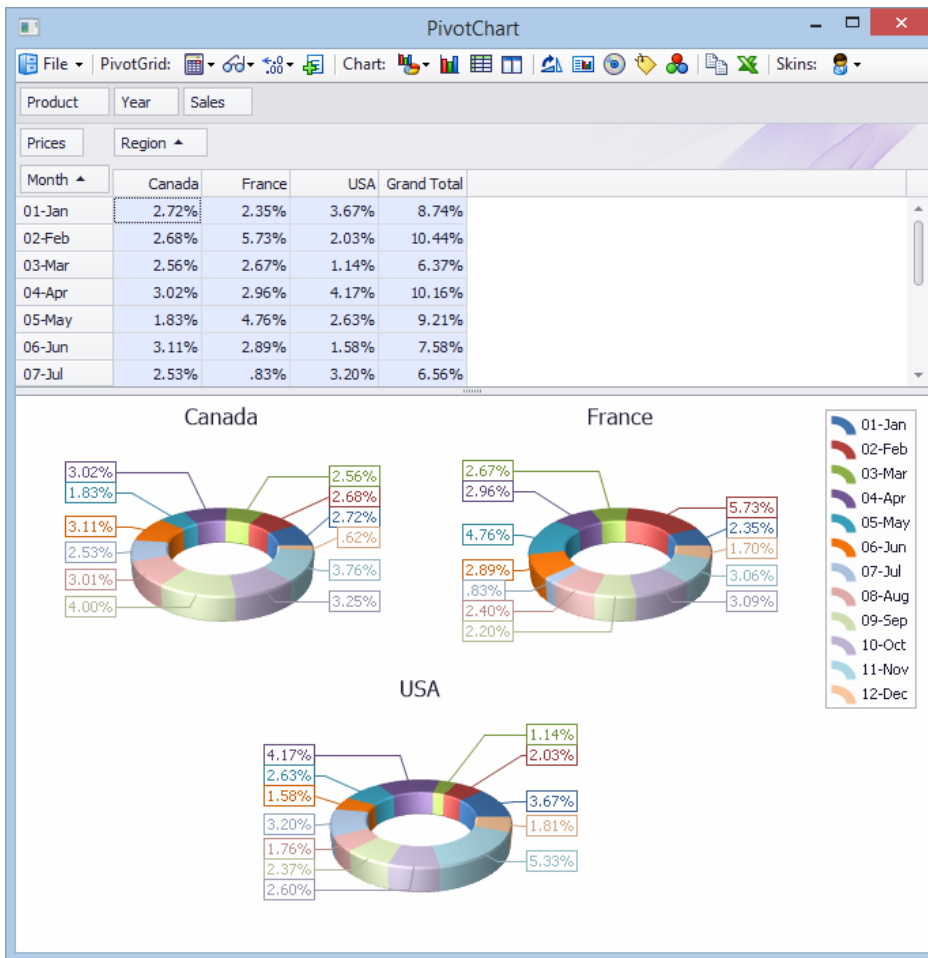
Now let's hide the legend:

```
'pc'wi '*xIsShowLegend' 0
```

If you want to see maximum prices by Month and by Region, you can do:

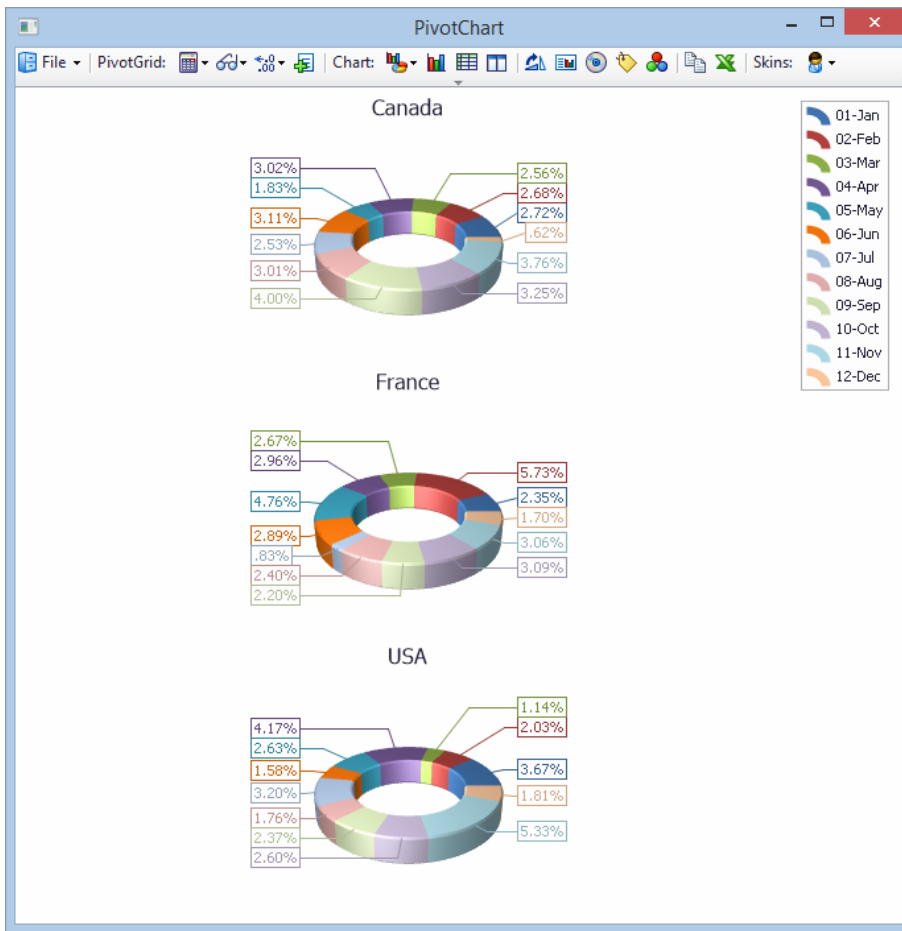
```
'pc'wi '*XSetArea' 'Region' 'col' 0
'pc'wi '*XSetArea' 'Year' 'filter' 0
'pc'wi '*XSetArea' 'Month' 'row' 0
'pc'wi '*XSetArea' 'Product' 'filter' 0
'pc'wi '*XSetDecimals' 'prices' 2
```

The **SetArea** method moves a field from an area to another area: the 3rd argument indicates what the position of the field should be in the specified area.



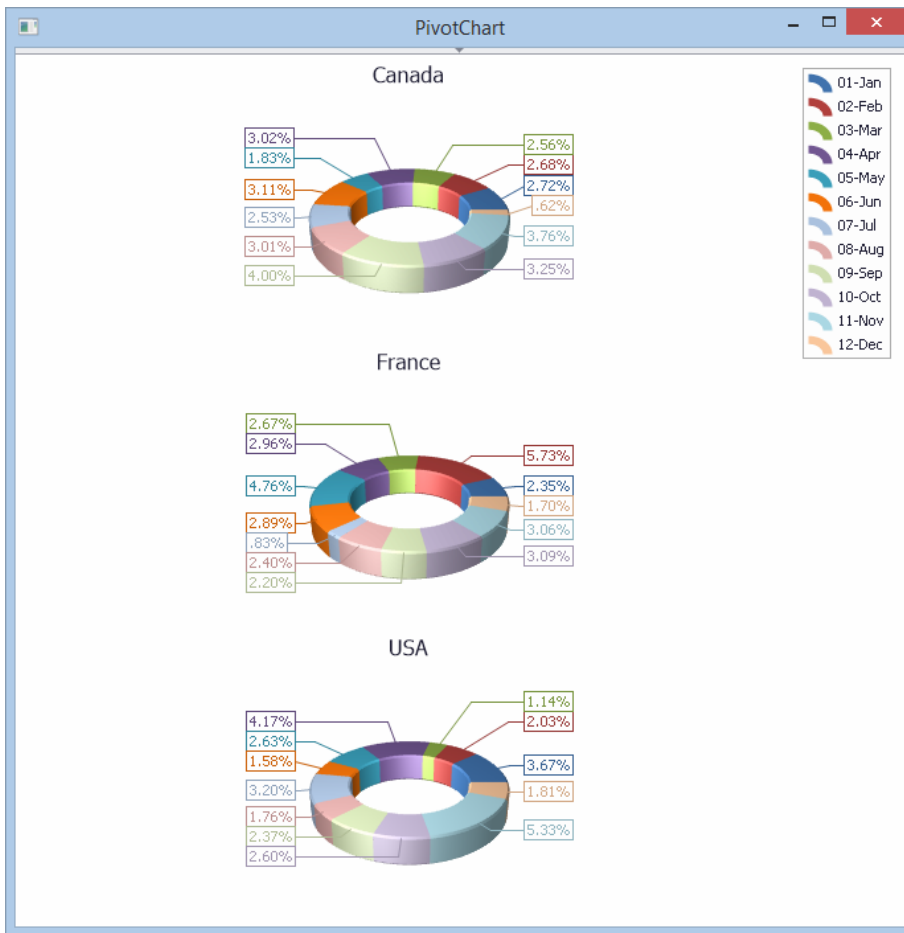
You can maximize the Chart or the Grid at any time using the **IsChartMaximized** or **isGridMaximized** method:

'pc'wi '*xIsChartMaximized'1



And you can hide the toolbar at any time using the `ToolbarVisible` property:

'pc'wi'*xToolbarVisible'0



Appendix 1

Here is the sample APL **PivotChart** function showing how to use **LC.PivotChart** and embed it in an APL+Win Form:

```
▽ PivotChart2;cntries;layout;months;nbRecs;products;years
[1]
[2]     nbRecs←300
[3]     months←'Jan' 'Feb' 'Mar' 'Apr' 'May' 'Jun' 'Jul' 'Aug' 'Sep' 'Oct' 'Nov' 'Dec'
[4]     months←(⌈2↑''0',⌈⌈''12),⌈''-',⌈''months
[5]     cntries←'France' 'Canada' 'USA'
[6]     products←'APL+Win' 'C#' 'VB.Net' 'Javascript' 'AngularJS'
[7]     years←2010+⌈5
[8]
[9]     A Define the PivotChart fields
[10]    A layout: 9-column nested array with one PivtoChart field per row
[11]    A [;1] = column name
[12]    A [;2] = column data (1D array)
[13]    A [;3] = area (filter, row, col or data)
[14]    A [;4] = summary type (average, count, custom, max, min, stddev, stddevp,
[15]    A         sum, var, varp)
[16]    A [;5] = summary display type (absvar, default or "", index, %col, %colgt, %gt,
[17]    A         %row, %rowgt, %var, rankcoll2s, rankcols2l, rankrowl2s, rankrows2l)
[18]    A [;6] = column width
[19]    A [;7] = % format string (i.e. "p2")
[20]    A [;8] = format string (i.e. "#,###.00")
[21]    A [;9] = sort order ("", "ascending" or "descending")
[22]
[23]    layout←6 9p∘''
[24]    layout[;1]←'Product' 'Year' 'Prices' 'Region' 'Sales' 'Month' A field names
[25]    layout[1;2]←∘products[?nbRecsppproducts] A data
[26]    layout[2;2]←∘years[?nbRecspyears]
[27]    layout[3;2]←∘?nbRecsp1000
[28]    layout[4;2]←∘cntries[?nbRecspcntries]
[29]    layout[5;2]←∘nbRecsp10
[30]    layout[6;2]←∘months[?nbRecspmonths]
[31]    layout[;3]←'row' 'col' 'data' 'filter' 'filter' 'filter' A field areas
[32]    A layout[;3]←'row' 'col' 'data' 'filter' 'data' 'col' A field areas
[33]    layout[;4]←'' '' 'average' '' 'sum' '' A summary types
```

```

[34] layout[;5]← ' ' 'values' ' ' '%gt' ' ' A summary display types
[35] layout[;6]←-1 A width (-1=autofit)
[36] layout[;7]←' ' ' "#.##%;-#.##%;'" ' " "#.##%;-#.##%;'" ' A pct format string
[37] layout[;8]←' ' ' "#.0 €;-#.0 €;" ' ' '#,##0' ' ' A format string
[38] layout[;9]←'descending' ' ' ' ' ' ' ' ' A sort
[39]
[40] ←'ff'□wi '*Create' 'zForm'('size'.6 .6)('*caption' 'zPivotChart Example')
[41] ←'ff'□wi '*pc.Create' 'zPivotChart'('where|c'0 0 '>>' '>>')('anchor' 'lrbt')('Populate'layout)
[42] ←'ff'□wi 'CenterScreen'
[43] ←'ff'□wi 'Show'
[44]

```

▽