

## RadBulletGraph For Silverlight and WPF

This tutorial will introduce the RadBulletGraph control, part of the Telerik suite of XAML controls. RadBulletGraph provides you with a quick overview of your data, offering an easy way to see how a featured value looks compared to a second value. RadBulletGraph also allows you to place a projection to get a full view of your data.

### Setting Up The Project

To begin, open Visual Studio and click on the Telerik menu option. Under *Rad Controls For Silverlight* click on *Create New Telerik Project*. Name your project, accept Silverlight 5 and in the Project Configuration Wizard dialog check *DataVisualization* (notice that the dependent references are automatically checked as well), as shown in figure 1.

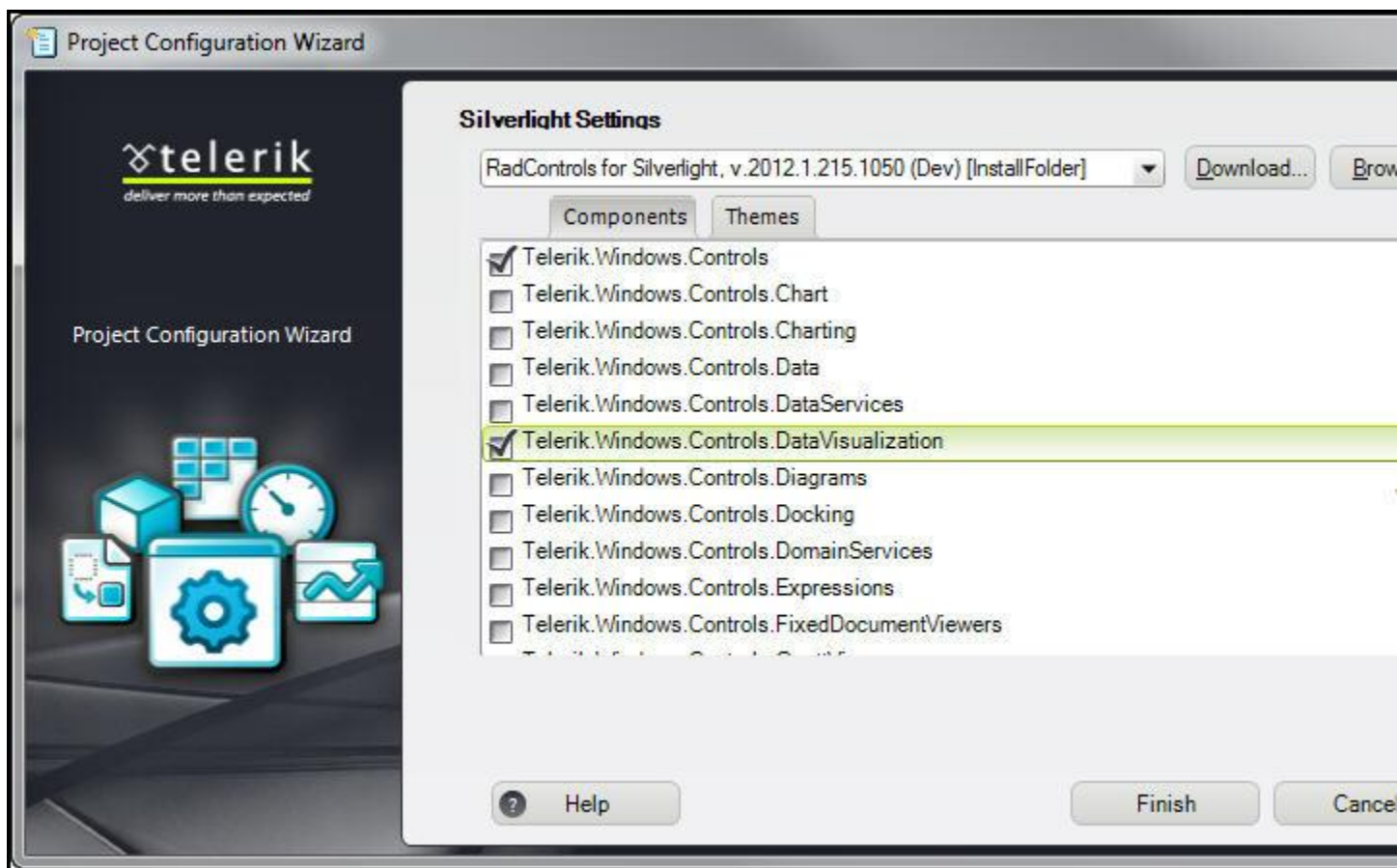


Figure 1

When you click ok, the necessary assemblies are added to the References as shown in figure 2

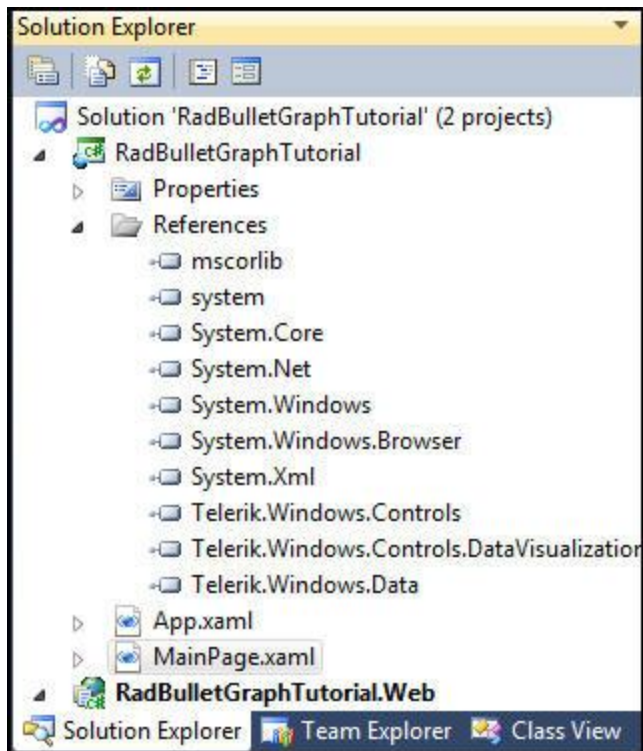


Figure 2

Your application will open to `MainPage.xaml` and, thanks to the Telerik Visual Studio extensions, the namespace `telerik` will already have been created in the XAML heading.

```
<UserControl x:Class="RadBulletGraphTutorial.MainPage"
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
    xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
    xmlns:telerik="http://schemas.telerik.com/2008/xaml/presentation"
    mc:Ignorable="d" d:DesignWidth="640" d:DesignHeight="480">
```

Begin by defining two columns in the Grid,

```
<Grid.ColumnDefinitions>
    <ColumnDefinition />
    <ColumnDefinition />
</Grid.ColumnDefinitions>
```

Add a `RadHorizontalBulletGraph`, setting its height and margin and then its key values:

```
<telerik:RadHorizontalBulletGraph Height="50"
    Margin="10"
    Minimum="0"
    Maximum="100"
    FeaturedMeasure="40">
```

```
ComparativeMeasure="75"
ProjectedValue="90"/>
```

Minimum and Maximum establish the extent of the scale. Featured measure is what we're comparing against the Comparative measure, and the projected value is the value we expect to end up at. As soon as you add this to the XAML you see the graph in the designer, as shown in figure 3,

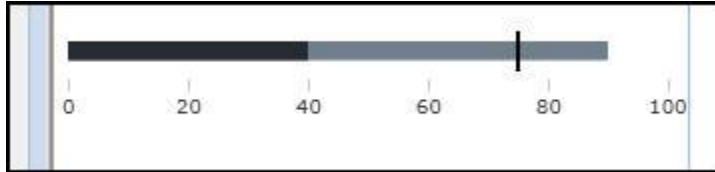


Figure 3

We can now add Qualitative Ranges. `<telerik:RadHorizontalBulletGraph.QualitativeRanges>`  
`<telerik:QualitativeRangeCollection>`  
`<telerik:QualitativeRange Brush="DarkGray"`  
`Value="50" />`  
`</telerik:QualitativeRangeCollection>`  
`</telerik:RadHorizontalBulletGraph.QualitativeRanges>`

Adding this first QualitativeRange adds a DarkGray representation of the value 50, as shown in figure 4,

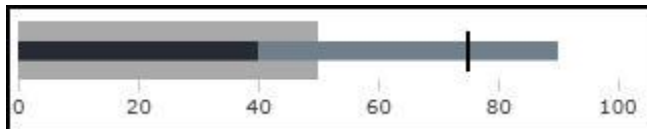


Figure 4

This is typically more meaningful when you add more than one Qualitative Range,

```
<telerik:QualitativeRangeCollection>
  <telerik:QualitativeRange Brush="DarkGray"
    Value="50" />
  <telerik:QualitativeRange Brush="LightGray"
    Value="75" />
  <telerik:QualitativeRange Brush="LightGreen"
    Value="100" />
</telerik:QualitativeRangeCollection>
```

When we run this we see the ranges superimposed on the other values, as shown in figure 5,

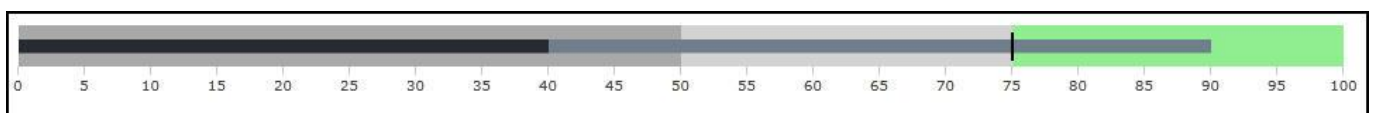


Figure 5

Note that if you shrink down the browser, the graph will adjust its numbering accordingly, as shown in figure 6,

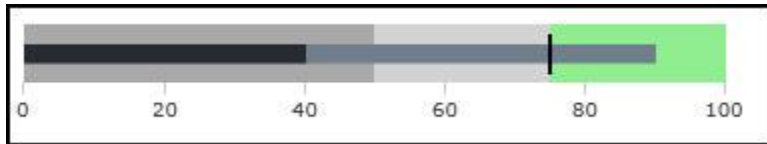


Figure 6

If we copy and paste the HorizontalBulletGraph and change the column to 1, and the type to VerticalBulletGraph, and we change Height=50 to Width=75, we get a second graph that looks like figure 7,

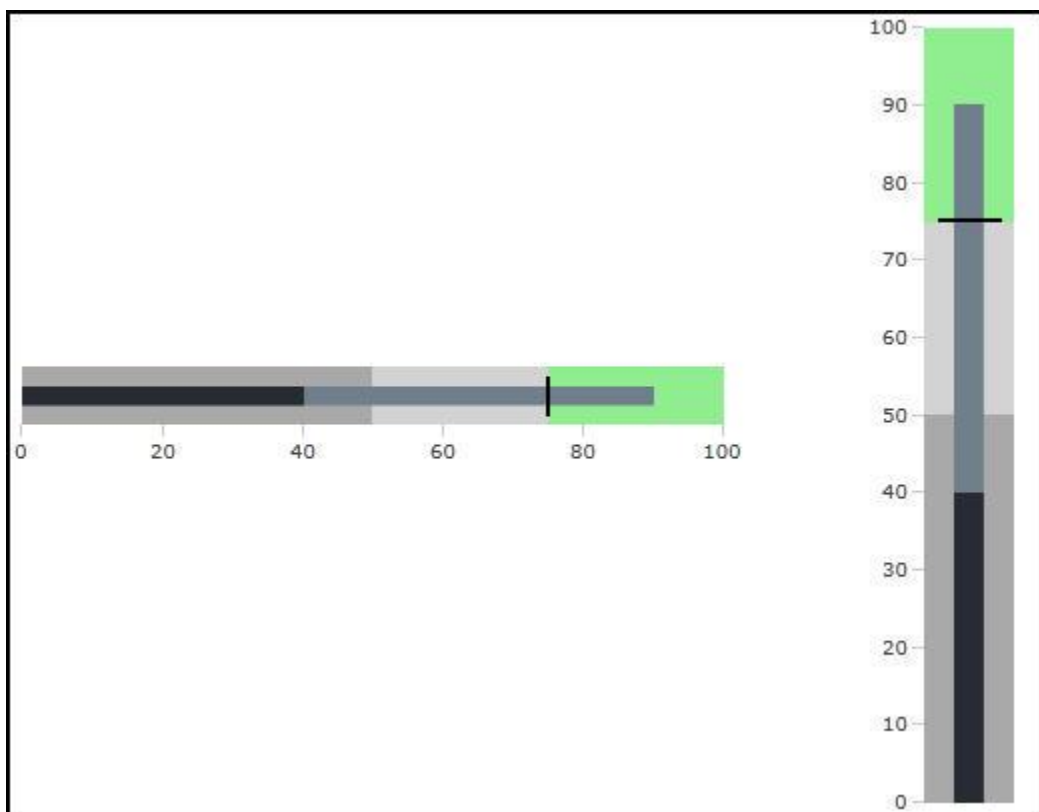


Figure 7

For completeness, here's the XAML for the vertical graph,

```
<telerik:RadVerticalBulletGraph Width="75"
    Grid.Column="1"
    Margin="10"
    Minimum="0"
    Maximum="100"
    FeaturedMeasure="40"
    ComparativeMeasure="75"
    ProjectedValue="90">
```

## Customizing Bullet Graphs

To get a good idea of the kinds of customization you can accomplish with the RadBulletGraph, create a new project, just as you did above.

Add a HorizontalBulletGraph with reasonable starting values,

```
<telerik:RadHorizontalBulletGraph Height="50"
    Margin="10"
    Minimum="0"
    Maximum="100"
    FeaturedMeasure="50"
    ComparativeMeasure="80"
    ProjectedValue="90">
```

Let's also add some ranges as this is a second area for customization,

```
<telerik:RadHorizontalBulletGraph.QualitativeRanges>
    <telerik:QualitativeRange Brush="PaleVioletRed"
        Value="33" />
    <telerik:QualitativeRange Brush="Yellow"
        Value="66" />
    <telerik:QualitativeRange Brush="LightGreen"
        Value="100" />
</telerik:RadHorizontalBulletGraph.QualitativeRanges>
```

The result can be seen when you run the application, as shown in figure 8,

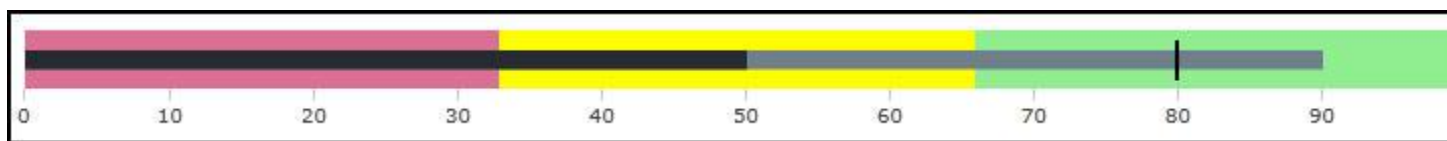


Figure 8

An alternative to using QualitativeRange Brushes, is to use Brushes within the Featured, Comparative and Projected values themselves. To see this, we'll comment out the QualitativeRanges and add brushes as follows,

```
<telerik:RadHorizontalBulletGraph Height="50"
    Margin="10"
    Minimum="0"
    Maximum="100"
    FeaturedMeasure="50"
    FeaturedMeasureBrush="Green"
    ComparativeMeasure="80">
```

```

ComparativeMeasureThickness="5"
ComparativeMeasureBrush="Red"
ProjectedValue="90"
ProjectedValueBrush="Orange">

```

The result of this is strikingly different, as shown in figure 9,



Figure 9

You can of course combine both effects.

You can also add additional comparative measures. To do so, just add the following,

```

<telerik:RadHorizontalBulletGraph Height="50"
    Margin="10"
    Minimum="0"
    Maximum="100"
    FeaturedMeasure="50"
    FeaturedMeasureBrush="Green"
    ComparativeMeasure="80"
    ComparativeMeasureBrush="Red"
    ComparativeMeasureThickness="5"
    ProjectedValue="90"
    ProjectedValueBrush="Orange">

    <telerik:RadHorizontalBulletGraph.AdditionalComparativeMeasures>
        <telerik:ComparativeMeasure Brush="Blue"
            Value="40"
            Thickness="5" />
    </telerik:RadHorizontalBulletGraph.AdditionalComparativeMeasures>
</telerik:RadHorizontalBulletGraph>

```

The result when you run the application is to have two comparative measures which, depending on your application, can be very useful. This is shown in figure 10,

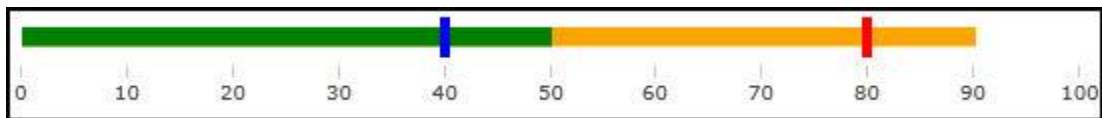


Figure 10

## Integrating With RadGridView

Start a new project as above, making sure that DataVisualization and GridView are both selected in the Project Configuration Wizard, as shown in figure 11,

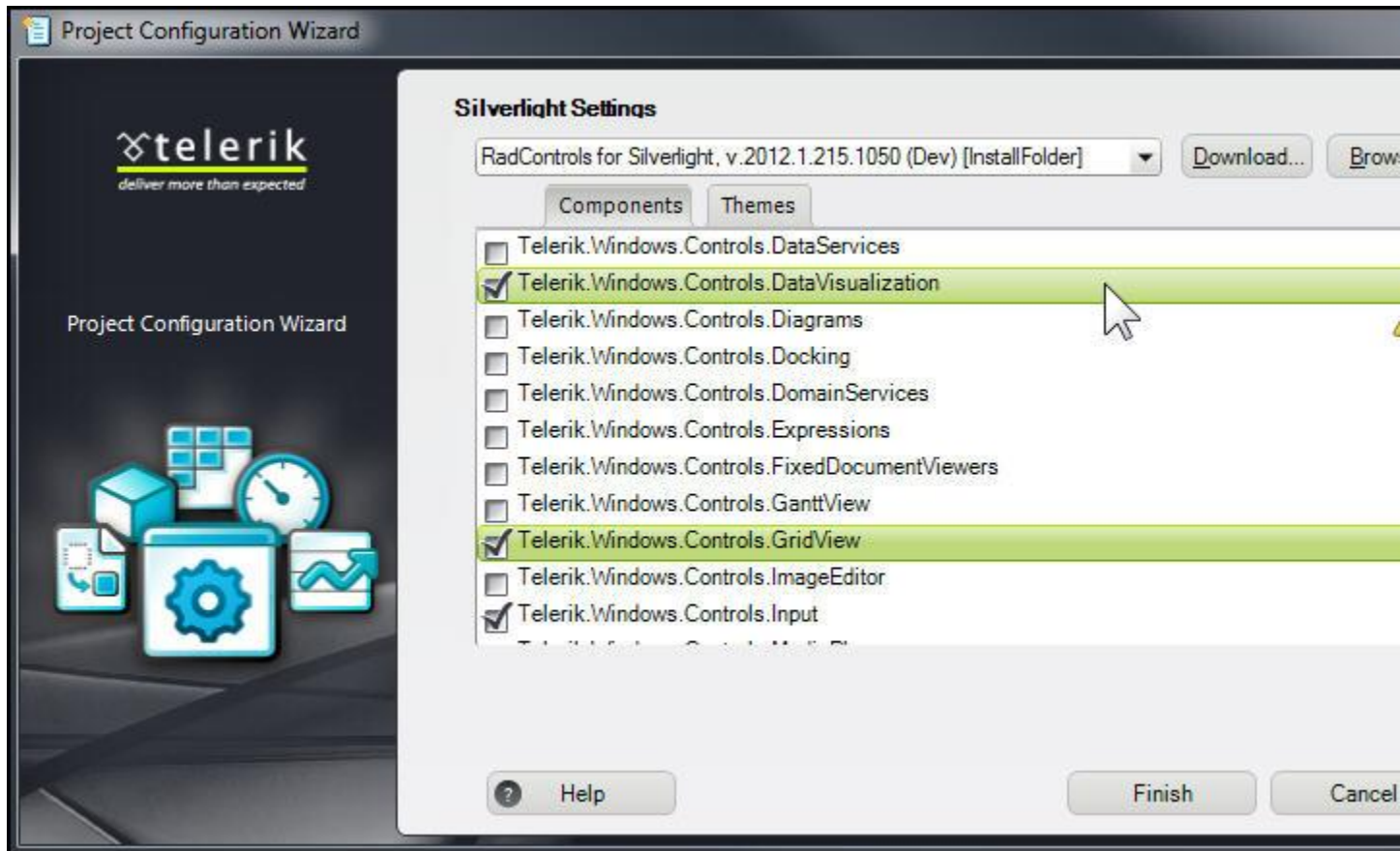


Figure 11

In the xaml all we're going to do is add a RadGridView,

```
<telerik:RadGridView Name="xRadGridView">
</telerik:RadGridView>
```

Open MainPage.xaml.cs and create a new class,

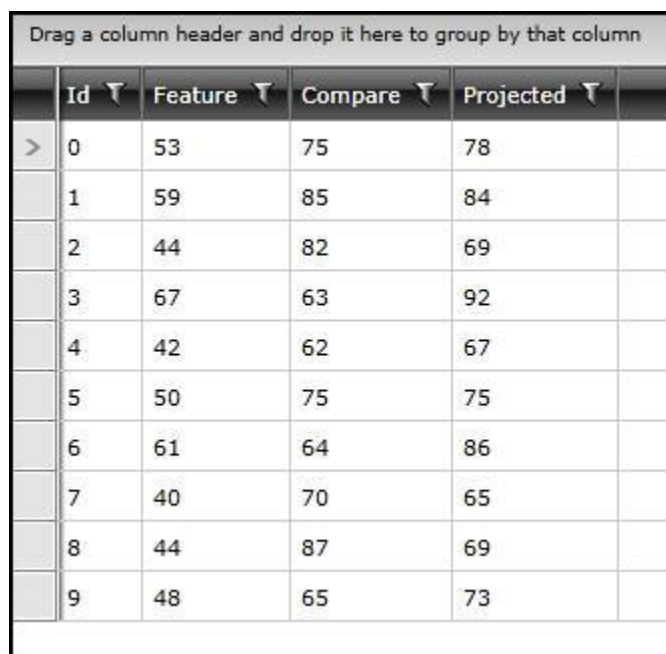
```
public class GridSampleItem
{
    public int Id { get; set; }
    public int Feature { get; set; }
    public int Compare { get; set; }
    public int Projected { get; set; }
}
```

We are now ready to programmatically create instances of `GridSampleItem` and add them to a collection, setting that collection as the `ItemsSource` for our `RadGridView`,

```
void MainPage_Loaded(object sender, RoutedEventArgs e)
{
    List<GridSampleItem> gridSamples = new List<GridSampleItem>();
    Random rnd = new Random()
    for (int i = 0; i < 10; i++)
    {
        var gsi = new GridSampleItem();
        gsi.Id = i;
        gsi.Feature = rnd.Next(40, 70);
        gsi.Compare = rnd.Next(60, 90);
        gsi.Projected = gsi.Feature + 25;
        gridSamples.Add(gsi);
    }

    xRadGridView.ItemsSource = gridSamples;
}
```

When we run this, `RadGridView` will provide a tidy look at our data, as shown in figure 12,



Drag a column header and drop it here to group by that column				
	Id	Feature	Compare	Projected
>	0	53	75	78
	1	59	85	84
	2	44	82	69
	3	67	63	92
	4	42	62	67
	5	50	75	75
	6	61	64	86
	7	40	70	65
	8	44	87	69
	9	48	65	73

Figure 12

This is very tidy, but it does not scale cognitively. That is, if we get up to 100 rows (let alone 10,000 rows!) it becomes very difficult to make meaningful comparisons.

We can make some dramatic improvements to this `GridView`, and we won't be doing it in code; instead turn to `MainPage.xaml`,



Let's return to the RadGridView and set AutoFormatColumns to false. We can then add two columns, one for ID and one we'll name Graph. We'll set the Graph column to read only as we'll have various values and it won't be meaningful to change the value of the graph (though there is a work around this that we will cover in an advanced video).

```
<telerik:RadGridView Name="xRadGridView" AutoGenerateColumns="False">
  <telerik:RadGridView.Columns>
    <telerik:GridViewDataColumn Header="ID"
                                DataMemberBinding="{Binding Id}" />
    <telerik:GridViewDataColumn Header="Graph" Width="200"
                                IsReadOnly="True">
      <telerik:GridViewDataColumn.CellTemplate>
        <DataTemplate>
          <telerik:RadHorizontalBulletGraph Minimum="0"
                                            MaxHeight="100"
                                            Height="40"
                                            Margin="10,3"
                                            FeaturedMeasure="{Binding Feature}"
                                            ComparativeMeasure="{Binding
Compare}"
                                            ProjectedValue="{Binding
Projected}" />
        </DataTemplate>
      </telerik:GridViewDataColumn.CellTemplate>
    </telerik:GridViewDataColumn>
  </telerik:RadGridView.Columns>
```

When we run this the ten rows are displayed with RadBulletGraphs, making the data much easier to apprehend, as seen in figure 13,

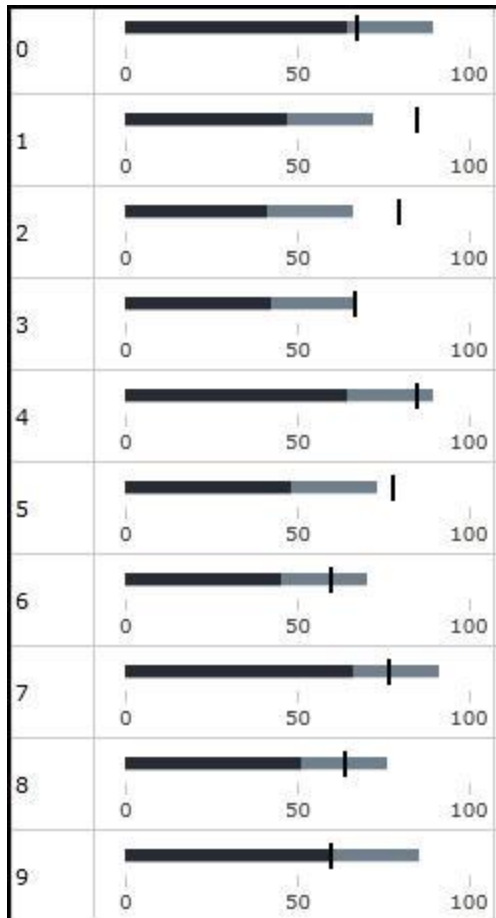


Figure 13

If you find this view does not give you sufficient precision, you can grab the right side of the column and drag it to the right, expanding the column and enhancing the precision, as shown in figure 14,

