

Working with Chart Multiple Axes – Nevron Chart Axis Docking

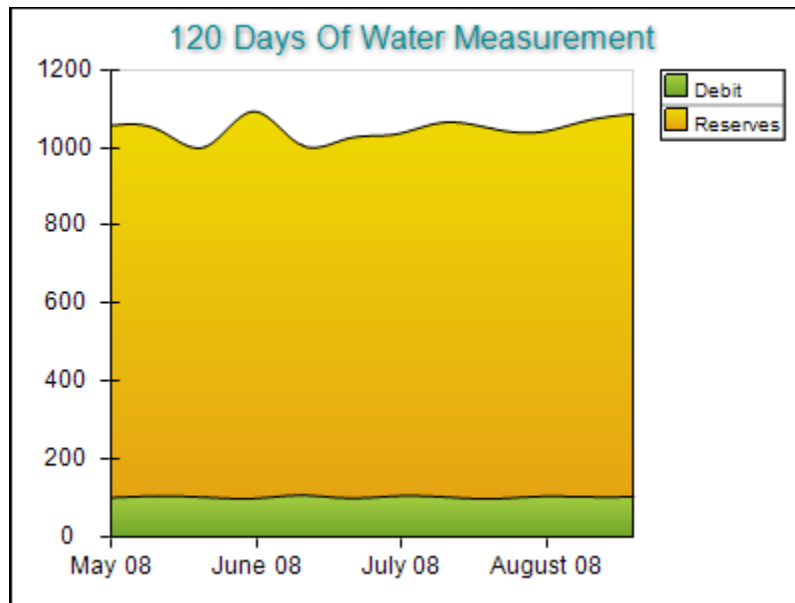


Introduction

Having a chart with several Y axes is a requirement you will meet very often when you develop charting enabled applications. It is essential that a good charting product supports flexible Y axis positioning and layout. This white paper outlines the different options you have with Nevron Chart for .NET to address a variety of situations requiring you to use multiple Y axes.

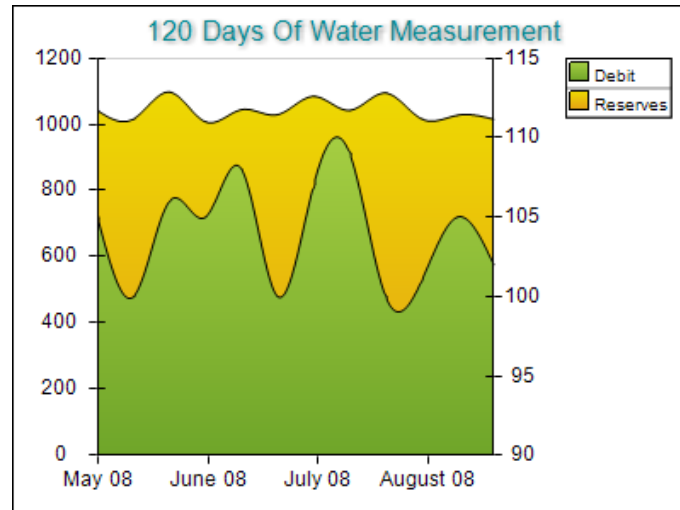
Features

The most common requirement is to have two Y axes and two series scaling on each of them. This is common when you need to compare two series of data that differ in magnitude or measurement, but you still want to have the ability to compare them visually in order to give the end user the ability to analyze trends or relations between the two. For example consider the following picture showing 120 days of imaginary water measurement:



It is clear that the user cannot properly analyze the two since water reserves differ in magnitude from water debit.

One solution would be to use a secondary axis for the debit data in order to show the two trends simultaneously:



On the above picture the debit measurement is shown on the secondary primary axis and is additionally rebased at 90. The code used to scale the debit area on the secondary Y axis is:

[C#]

```
NAxis secondaryY = (NAxis)chart.Axis(StandardAxis.SecondaryY);
secondaryY.Visible = true;
```

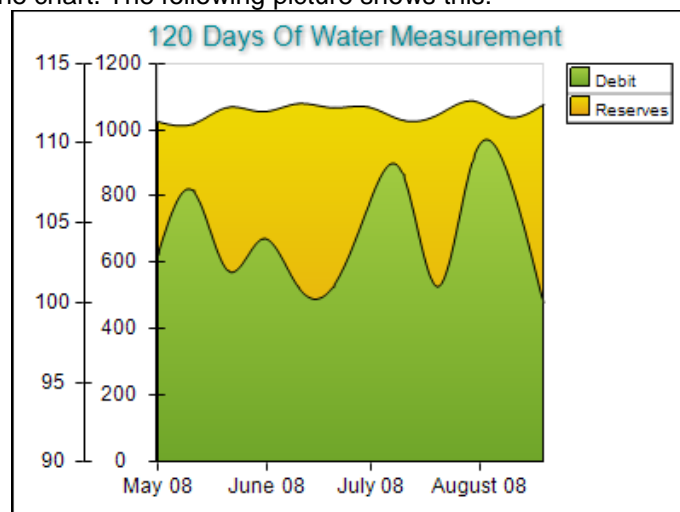
```
areaDebit.DisplayOnAxis(StandardAxis.PrimaryY, false);
areaDebit.DisplayOnAxis(StandardAxis.SecondaryY, true);
```

[VB.NET]

```
Dim secondaryY As NAxis = chart.Axis(StandardAxis.SecondaryY)
secondaryY.Visible = True
```

```
areaDebit.DisplayOnAxis(StandardAxis.PrimaryY, False)
areaDebit.DisplayOnAxis(StandardAxis.SecondaryY, True)
```

Note that the chart area has shrunk when the secondary Y axis is made visible. In certain cases you may want to have both axes on the same side of the chart. The following picture shows this:



In order to change the position of the SecondaryY axes you must modify its anchor:

[C#]

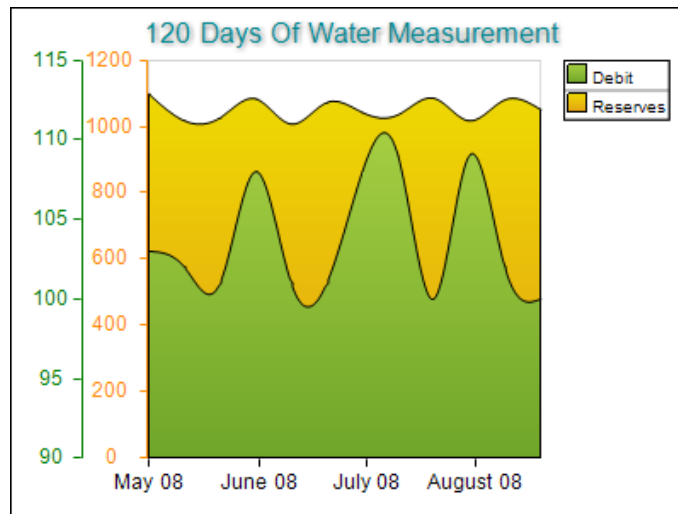
```
NAxis secondaryY = (NAxis)chart.Axis(StandardAxis.SecondaryY);
secondaryY.Visible = true;
secondaryY.Anchor = new
NDockAxisAnchor(AxisDockZone.FrontLeft);
```

[VB.NET]

```
Dim secondaryY As NAxis = chart.Axis(StandardAxis.SecondaryY)
secondaryY.Visible = True
secondaryY.Anchor = New
NDockAxisAnchor(AxisDockZone.FrontLeft)
```

Now the secondary Y axis is docked on the left side to the primary Y (showing the range [0, 1200]).

The above chart needs another touch. When you dock several axes to a single axis dock zone (in the above case left zone) you may want to introduce some color coding linking the axes to the series that scale on them. This is necessary because the user cannot visually comprehend whether the debit scales from 90 to 115 or from 0 to 1200. The following picture shows how the axes are now color coded to match the content that scales on them:



Note that we also disabled the inner major ticks in the axes in order to save some space. This is achieved through the following code:

[C#]

```
NAxis primaryY = (NAxis)chart.Axis(StandardAxis.PrimaryY);
NStandardScaleConfigurator primaryScaleY = primaryY.ScaleConfigurator as
NStandardScaleConfigurator;
primaryScaleY.RulerStyle.BorderStyle.Color = Color.DarkOrange;
primaryScaleY.OuterMajorTickStyle.LineStyle.Color = Color.DarkOrange;
primaryScaleY.InnerMajorTickStyle.Length = new NLength(0);
primaryScaleY.LabelStyle.TextStyle.FillStyle = new NColorFillStyle(Color.DarkOrange);

NAxis secondaryY = (NAxis)chart.Axis(StandardAxis.SecondaryY);
NLinearScaleConfigurator secondaryScaleY = secondaryY.ScaleConfigurator as
NLinearScaleConfigurator;
secondaryScaleY.RulerStyle.BorderStyle.Color = Color.Green;
secondaryScaleY.OuterMajorTickStyle.LineStyle.Color = Color.Green;
secondaryScaleY.InnerMajorTickStyle.Length = new NLength(0);
secondaryScaleY.UseOrigin = true;
secondaryScaleY.Origin = 100;
secondaryScaleY.LabelStyle.TextStyle.FillStyle = new NColorFillStyle(Color.Green);
```

Visualize Your Success

[VB.NET]

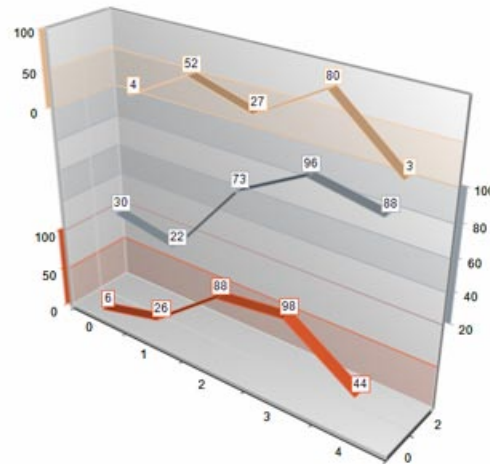
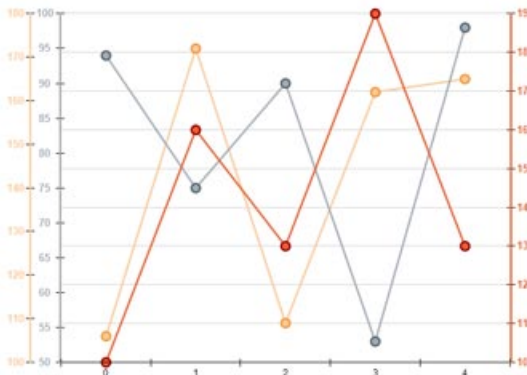
```

Dim primaryY As NAxis = chart.Axis(StandardAxis.PrimaryY)
Dim primaryScaleY As NStandardScaleConfigurator = CType(primaryY.ScaleConfigurator,
NStandardScaleConfigurator)
primaryScaleY.RulerStyle.BorderStyle.Color = Color.DarkOrange
primaryScaleY.OuterMajorTickStyle.LineStyle.Color = Color.DarkOrange
primaryScaleY.InnerMajorTickStyle.Length = New NLength(0)
primaryScaleY.LabelStyle.TextStyle.FillStyle = New NColorFillStyle(Color.DarkOrange)

Dim secondaryY As NAxis = chart.Axis(StandardAxis.SecondaryY)
Dim secondaryScaleY As NLinearScaleConfigurator = CType(secondaryY.ScaleConfigurator,
NLinearScaleConfigurator)
secondaryScaleY.RulerStyle.BorderStyle.Color = Color.Green
secondaryScaleY.OuterMajorTickStyle.LineStyle.Color = Color.Green
secondaryScaleY.InnerMajorTickStyle.Length = New NLength(0)
secondaryScaleY.UseOrigin = True
secondaryScaleY.Origin = 100
secondaryScaleY.LabelStyle.TextStyle.FillStyle = New NColorFillStyle(Color.Green)

```

Nevron Chart can have an unlimited number of custom vertical and horizontal axes. You can specify the axis on which a particular series is scaled. When using axis docking you can also specify whether the axis will create a separate level in the dock zone or use the last created one. This allows you to have two or more axes in a zone level that share a percentage of the chart wall.



Conclusion

Nevron Chart for .NET fully supports charts with several X and Y axes and additionally allows you to customize a variety of options to get the desired result for your charts. Very few charting packages support docking several axes on the left or right side of the chart area. More information on axis positioning is available in the Users Guide shipped with the fully functional, not time restricted evaluation of the component available at <http://www.nevron.com>.



ISV/Software Solutions

