

Notes on Windows Forms Chart in WPF

Windows Forms has a charting tool which is not available in WPF unless you add a toolbox. It may be easier in many cases to add the code that is necessary to use Window Forms in WPF. This example shows how to add the Windows Forms Chart tool to a WPF project and plot a sinusoid.

1. Create a new WPF project. I named mine *WPFFrmChtEx2*. Use either the XAML or the properties for the main window to set the size to 1024 x 768. Set the Window start up screen to "CenterScreen". The window will have a default grid. Use the XAML or properties to give this grid a name. I used *grdFrmCht*. Add a *Canvas* to your project along with a button. I named the canvas *cnvChart* and the button *btnStart*. You will also need a canvas loaded event and a button click event. My XAML code looks like this:

```
<Window x:Class="WPFFrmChtEx2.MainWindow"
        xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
        xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
        Title="MainWindow" Height="768" Width="1024"
        WindowStartupLocation="CenterScreen">
    <Grid Name="grdFrmCht">
        <Canvas x:Name="cnvChart" HorizontalAlignment="Left" Height="590"
            Margin="10,10,0,0" VerticalAlignment="Top" Width="982"
            Loaded="cnvChart_Loaded"/>
        <Button x:Name="btnStart" Content="Plot" HorizontalAlignment="Left"
            Height="55" Margin="415,642,0,0" VerticalAlignment="Top"
            Width="170" FontSize="24" Click="Button_Click"/>
    </Grid>
</Window>
```

2. You will need to add four references. Right click on the project name in the Solution Explorer and select Add → Reference. Under the Framework tab add references to
 System.Windows.Forms
 System.Windows.Forms.DataVisualization
 WindowsFormsIntegration
 System.Drawing

3. Add the following four *using* directives to the .cs file code:

```
using System.Windows.Forms;
using System.Windows.Forms.Integration;
using System.Windows.Forms.DataVisualization.Charting;
using System.Drawing;
```

4. Declare a chart as a private variable of the class like this:

```
private Chart chtSin = new Chart();
```

5. We will add the Forms chart tool programmatically. In the *cnvChart_Loaded* event add the following lines:

```
{System.Windows.Forms.Integration.WindowsFormsHost host =
    new System.Windows.Forms.Integration.WindowsFormsHost();
```

```

host.Child = chtSin;
// Add the chart to the canvas so it can be displayed.
this.cnvChart.Children.Add(host);
}

```

6. Finally we write the button click event. This sets up the chart. In this case the series plots a sinusoid.

```

private void Button_Click(object sender, RoutedEventArgs e)
{
    double t, y, tIncr;
    chtSin.ChartAreas.Add("Default");
    // Add a series with some data points.
    chtSin.Width = 900;
    chtSin.Height = 500;
    chtSin.Location = new System.Drawing.Point(10, 10);
    Series sinSeries = new Series();
    sinSeries.ChartType = SeriesChartType.Line;
    tIncr = 4*Math.PI/500;
    for(t=0;t<4*Math.PI;t+=tIncr)
        {y = Math.Sin(2*Math.PI*t);
        sinSeries.Points.AddXY(t, y);
        }
    chtSin.Series.Add(sinSeries);
    chtSin.ChartAreas[0].AxisX.Title = "Time";
    chtSin.ChartAreas[0].AxisX.LabelStyle.Format = "{0.00}";
    chtSin.ChartAreas[0].AxisY.Title = "Volts";
    chtSin.ChartAreas[0].AxisY.LabelStyle.Format = "{0.00}";
}

```

The complete project can be downloaded from the web site as WPFFormsChartEx2.zip.

